

Manual of PATENT EXAMINING PROCEDURE

Original Eighth Edition, August 2001
Latest Revision May 2004



U.S. DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

Rev. 2, May 2004

The U.S. Patent and Trademark Office does not handle the sale of the Manual, distribution of notices and revisions, or change of address of those on the subscription list. Correspondence relating to existing subscriptions should be sent to the Superintendent of Documents at the following address:

Superintendent of Documents
Mail List Section
Washington, DC 20402
Telephone: 202-512-2267

Inquiries relating to purchasing the Manual should be directed to:

Superintendent of Documents
United States Government Printing Office
Washington, DC 20402
Telephone: 202-512-1800

Orders for reproduced copies of individual replacement pages or of previous revisions of the Manual should be sent to the following address:

Mail Stop Document Services
Director of the U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
Telephone: 1-800-972-6382 or 703-308-9726

Previous editions and revisions of the Manual are available on microfilm in the Patent Search Room.
The Manual is available on CD-ROM and on diskette from:

U.S. Patent and Trademark Office
Office of Electronic Information Products and Services
P.O. Box 1450
Alexandria, VA 22313-1450
Telephone: 703-306-2600

Employees of the U.S. Patent and Trademark Office should direct their requests for the Manual, replacement pages, notices, and revisions to the Patent Academy.
Telephone: 703-308-9660

Pursuant to the Patent and Trademark Office Efficiency Act (PTOEA) (Pub. L. 106-113, 113 Stat. 1501A-572), the head of the United States Patent and Trademark Office (USPTO) is the "Under Secretary of Commerce for Intellectual Property and Director of the United States Patent and Trademark Office." The Director is assisted by the "Deputy Under Secretary of Commerce for Intellectual Property and Deputy Director of the United States Patent and Trademark Office." The patent operations of the USPTO are now headed by the "Commissioner for Patents." The trademark operations of the USPTO are now headed by the "Commissioner for Trademarks." Under section 4741(b) of the PTOEA, any reference to the Commissioner of Patents and Trademarks, the Assistant Commissioner for Patents, or the Assistant Commissioner for Trademarks is deemed to refer to the Director, the Commissioner for Patents, or the Commissioner for Trademarks, respectively. See "Reestablishment of the Patent and Trademark Office as the United States Patent and Trademark Office" published in the *Federal Register* at 65 FR 17858 (Apr. 5, 2000), and in the *Official Gazette of the United States Patent and Trademark Office* at 1234 O.G. 41 (May 9, 2000).

Additions to the text of the Manual are indicated by arrows (><) inserted in the text. Deletions are indicated by a single asterisk (*) where a single word was deleted and by two asterisks (**) where more than one word was deleted. The use of three or five asterisks in the body of the laws, rules, treaties, and administrative instructions indicates a portion of the law, rule, treaty, or administrative instruction which was not reproduced.

First Edition, November 1949
Second Edition, November 1953
Third Edition, November 1961
Fourth Edition, June 1979
Fifth Edition, August 1983
Sixth Edition, January 1995
Seventh Edition, July 1998
Eighth Edition, August 2001
Revision 1, February 2003
Revision 2, May 2004

Rev. 2, May 2004

608.01(p)

MANUAL OF PATENT EXAMINING PROCEDURE

comply with 37 CFR 1.75(d)(1), applicant will be required to make appropriate amendment to the description to provide clear support or antecedent basis for the terms appearing in the claims provided no new matter is introduced.<

The specification should be objected to if it does not provide proper antecedent basis for the claims by using form paragraph 7.44.

¶ 7.44 Claimed Subject Matter Not in Specification

The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: [1]

608.01(p) Completeness [R-2]

Newly filed applications obviously failing to disclose an invention with the clarity required are discussed in MPEP § 702.01.

A disclosure in an application, to be complete, must contain such description and details as to enable any person skilled in the art or science to which the invention pertains to make and use the invention as of its filing date. *In re Glass*, 492 F.2d 1228, 181 USPQ 31 (CCPA 1974).

While the prior art setting may be mentioned in general terms, the essential novelty, the essence of the invention, must be described in such details, including proportions and techniques, where necessary, as to enable those persons skilled in the art to make and utilize the invention.

Specific operative embodiments or examples of the invention must be set forth. Examples and description should be of sufficient scope as to justify the scope of the claims. *Markush* claims must be provided with support in the disclosure for each member of the *Markush* group. Where the constitution and formula of a chemical compound is stated only as a probability or speculation, the disclosure is not sufficient to support claims identifying the compound by such composition or formula.

A complete disclosure should include a statement of utility. This usually presents no problem in mechanical cases. In chemical cases, varying degrees of specificity are required.

A disclosure involving a new chemical compound or composition must teach persons skilled in the art how to make the compound or composition. Incom-

plete teachings may not be completed by reference to subsequently filed applications.

For “Guidelines For Examination Of Applications For Compliance With The Utility Requirement of 35 U.S.C. 101,” see MPEP § 2107.

For “General Principles Governing Utility Rejections,” see MPEP § 2107.01.

For a discussion of the utility requirement under 35 U.S.C. 112, first paragraph, in drug cases, see MPEP § 2107.03 and § 2164.06(a).

For “Procedural Considerations Related to Rejections for Lack of Utility,” see MPEP § 2107.02.

For “Special Considerations for Asserted Therapeutic or Pharmacological Utilities,” see MPEP § 2107.03.

I. INCORPORATION BY REFERENCE

The *>Director< has considerable discretion in determining what may or may not be incorporated by reference in a patent application. *General Electric Co. v. Brenner*, 407 F.2d 1258, 159 USPQ 335 (D.C. Cir. 1968). The incorporation by reference practice with respect to applications which issue as U.S. patents provides the public with a patent disclosure which minimizes the public’s burden to search for and obtain copies of documents incorporated by reference which may not be readily available. Through the Office’s incorporation by reference policy, the Office ensures that reasonably complete disclosures are published as U.S. patents. The following is the manner in which the *>Director< has elected to exercise that discretion. Section A provides the guidance for incorporation by reference in applications which are to issue as U.S. patents. Section B provides guidance for incorporation by reference in benefit applications; i.e., those domestic (35 U.S.C. 120) or foreign (35 U.S.C. 119(a)) applications relied on to establish an earlier effective filing date. >See MPEP § 2181 for the impact of incorporation by reference on the determination of whether applicant has complied with the requirements of 35 U.S.C. 112, second paragraph when 35 U.S.C. 112, sixth paragraph is invoked.<

A. Review of Applications Which Are To Issue as Patents.

An application as filed must be complete in itself in order to comply with 35 U.S.C. 112. Material nevertheless may be incorporated by reference, *Ex parte*

PARTS, FORM, AND CONTENT OF APPLICATION

608.01(p)

*>*Schwarze*<, 151 USPQ 426 (Bd. Ape. 1966). An application for a patent when filed may incorporate “essential material” by reference to (1) a U.S. patent, (2) a U.S. patent application publication, or (3) a pending U.S. application, subject to the conditions set forth below.

“Essential material” is defined as that which is necessary to (1) describe the claimed invention, (2) provide an enabling disclosure of the claimed invention, or (3) describe the best mode (35 U.S.C. 112). In any application which is to issue as a U.S. patent, essential material may not be incorporated by reference to (1) patents or applications published by foreign countries or a regional patent office, (2) non-patent publications, (3) a U.S. patent or application which itself incorporates “essential material” by reference, or (4) a foreign application.

Nonessential subject matter may be incorporated by reference to (1) patents or applications published by the United States or foreign countries or regional patent offices, (2) prior filed, commonly owned U.S. applications, or (3) non-patent publications however, aperiens and/or other forms of browser executable code cannot be incorporated by reference. See MPEP § 608.01. Nonessential subject matter is subject matter referred to for purposes of indicating the background of the invention or illustrating the state of the art.

Mere reference to another application, patent, or publication is not an incorporation of anything therein into the application containing such reference for the purpose of the disclosure required by 35 U.S.C. 112, first paragraph. *In re de Seversky*, 474 F.2d 671, 177 USPQ 144 (CCPA 1973). In addition to other requirements for an application, the referencing application should include an identification of the referenced patent, application, or publication. Particular attention should be directed to specific portions of the referenced document where the subject matter being incorporated may be found. Guidelines for situations where applicant is permitted to fill in a number for Application No. _____ left blank in the application as filed can be found in *In re Fouché*, 439 F.2d 1237, 169 USPQ 429 (CCPA 1971) (Abandoned applications less than 20 years old can be incorporated by reference to the same extent as copending applications; both types are open to the public upon the referencing application issuing as a patent. See MPEP § 103).

1. Complete Disclosure Filed

If an application is filed with a complete disclosure, essential material may be canceled by amendment and may be substituted by reference to a U.S. patent or an earlier filed pending U.S. application. The amendment must be accompanied by an affidavit or declaration signed by the applicant, or a practitioner representing the applicant, stating that the material canceled from the application is the same material that has been incorporated by reference.

If an application as filed incorporates essential material by reference to a U.S. patent or a pending and commonly owned U.S. application, applicant may be required prior to examination to furnish the Office with a copy of the referenced material together with an affidavit or declaration executed by the applicant, or a practitioner representing the applicant, stating that the copy consists of the same material incorporated by reference in the referencing application. However, if a copy of a printed U.S. patent is furnished, no affidavit or declaration is required.

Prior to allowance of an application that incorporates essential material by reference to a pending U.S. application, the examiner shall determine if the referenced application has been published or issued as a patent. If the referenced application has been published or issued as a patent, the examiner shall enter the U.S. Patent Application Publication No. or the U.S. Patent No. of the referenced application in the specification of the referencing application (see MPEP § 1302.04). If the referenced application has not been published or issued as a patent, applicant will be required to amend the disclosure of the referencing application to include the material incorporated by reference. The amendment must be accompanied by an affidavit or declaration executed by the applicant, or a practitioner representing the applicant, stating the amendatory material consists of the same material incorporated by reference in the referencing application.

2. Improper Incorporation

The filing date of any application wherein essential material is improperly incorporated by reference to a foreign application or patent or to a publication will not be affected because of the reference. In such a case, the applicant will be required to amend the spec-

608.01(p)

MANUAL OF PATENT EXAMINING PROCEDURE

ification to include the material incorporated by reference. The following form paragraphs may be used.

¶ 6.19 *Incorporation by Reference, Foreign Patent or Application*

The incorporation of essential material in the specification by reference to a foreign application or patent, or to a publication is improper. Applicant is required to amend the disclosure to include the material incorporated by reference. The amendment must be accompanied by an affidavit or declaration executed by the applicant, or a practitioner representing the applicant, stating that the amendatory material consists of the same material incorporated by reference in the referencing application. *In re Hawkins*, 486 F.2d 569, 179 USPQ 157 (CCPA 1973); *In re Hawkins*, 486 F.2d 579, 179 USPQ 163 (CCPA 1973); *In re Hawkins*, 486 F.2d 577, 179 USPQ 167 (CCPA 1973).

¶ 6.19.01 *Improper Incorporation by Reference, General*

The attempt to incorporate subject matter into this application by reference to [1] is improper because [2].

Examiner Note:

1. In bracket 1, identify the document such as an application or patent number or other identification.
2. In bracket 2, give reason why it is improper.

The amendment must be accompanied by an affidavit or declaration executed by the applicant, or a practitioner representing the applicant, stating that the amendatory material consists of the same material incorporated by reference in the referencing application. *In re Hawkins*, 486 F.2d 569, 179 USPQ 157 (CCPA 1973); *In re Hawkins*, 486 F.2d 579, 179 USPQ 163 (CCPA 1973); *In re Hawkins*, 486 F.2d 577, 179 USPQ 167 (CCPA 1973).

Reliance on a commonly assigned copending application by a different inventor may ordinarily be made for the purpose of completing the disclosure. See *In re Fried*, 329 F.2d 323, 141 USPQ 27 (CCPA 1964), and *General Electric Co. v. Brenner*, 407 F.2d 1258, 159 USPQ 335 (D.C. Cir. 1968).

Since a disclosure must be complete as of the filing date, subsequent publications or subsequently filed applications cannot be relied on to establish a constructive reduction to practice or an enabling disclosure as of the filing date. *White Consol. Indus., Inc. v. Vega Servo-Control, Inc.*, 713 F.2d 788, 218 USPQ 961 (Fed. Cir. 1983); *In re Scarbrough*, 500 F.2d 560, 182 USPQ 298 (CCPA 1974); *In re Glass*, 492 F.2d 1228, 181 USPQ 31 (CCPA 1974).

B. Review of Applications Which Are Relied on To Establish an Earlier Effective Filing Date.

The limitations on the material which may be incorporated by reference in U.S. patent applications which are to issue as U.S. patents do not apply to applications relied on only to establish an earlier effective filing date under 35 U.S.C. 119 or 35 U.S.C. 120. Neither 35 U.S.C. 119(a) nor 35 U.S.C. 120 places any restrictions or limitations as to how the claimed invention must be disclosed in the earlier application to comply with 35 U.S.C. 112, first paragraph. Accordingly, an application is entitled to rely upon the filing date of an earlier application, even if the earlier application itself incorporates essential material by reference to another document. See *Ex parte Maziere*, 27 USPQ2d 1705, 1706-07 (Bd. Pat. App. & Inter. 1993).

The reason for incorporation by reference practice with respect to applications which are to issue as U.S. patents is to provide the public with a patent disclosure which minimizes the public's burden to search for and obtain copies of documents incorporated by reference which may not be readily available. Through the Office's incorporation by reference policy, the Office ensures that reasonably complete disclosures are published as U.S. patents. The same policy concern does not apply where the sole purpose for which an applicant relies on an earlier U.S. or foreign application is to establish an earlier filing date. Incorporation by reference in the earlier application of (1) patents or applications published by foreign countries or regional patent offices, (2) nonpatent publications, (3) a U.S. patent or application which itself incorporates "essential material" by reference, or (4) a foreign application, is not critical in the case of a "benefit" application.

When an applicant, or a patent owner in a reexamination or interference, claims the benefit of the filing date of an earlier application which incorporates material by reference, the applicant or patent owner may be required to supply copies of the material incorporated by reference. For example, an applicant may claim the benefit of the filing date of a foreign application which itself incorporates by reference another earlier filed foreign application. If necessary, due to an intervening reference, applicant should be required to supply a copy of the earlier filed foreign application, along with an English language translation.

PARTS, FORM, AND CONTENT OF APPLICATION

608.01(q)

tion. A review can then be made of the foreign application and all material incorporated by reference to determine whether the foreign application discloses the invention sought to be patented in the manner required by the first paragraph of 35 U.S.C. 112 so that benefit may be accorded. *In re Gosteli*, 872 F.2d 1008, 10 USPQ2d 1614 (Fed. Cir. 1989).

>As a safeguard against the omission of a portion of a prior application for which priority is claimed under 35 U.S.C. 119(a)-(d) or (f), or for which benefit is claimed under 35 U.S.C. 119(e) or 120, applicant may include a statement at the time of filing of the later application incorporating by reference the prior application. See MPEP § 201.06(c) where domestic benefit is claimed. See MPEP § 201.13 where foreign priority is claimed. The inclusion of such an incorporation by reference statement in the later-filed application will permit applicant to include subject matter from the prior application into the later-filed application without the subject matter being considered as new matter. For the incorporation by reference to be effective as a proper safeguard, the incorporation by reference statement must be filed at the time of filing of the later-filed application. An incorporation by reference statement added after an application's filing date is not effective because no new matter can be added to an application after its filing date (see 35 U.S.C. 132(a)).<

II. SIMULATED OR PREDICTED TEST RESULTS OR PROPHETIC EXAMPLES

Simulated or predicted test results and prophetic examples (paper examples) are permitted in patent applications. Working examples correspond to work actually performed and may describe tests which have actually been conducted and results that were achieved. Paper examples describe the manner and process of making an embodiment of the invention which has not actually been conducted. Paper examples should not be represented as work actually done. No results should be represented as actual results unless they have actually been achieved. Paper examples should not be described using the past tense. >*Hoffman-La Roche, Inc. v. Promega Corp.*, 323 F.3d 1354, 1367, 66 USPQ2d 1385, 1394 (Fed. Cir. 2003).<

For problems arising from the designation of materials by trademarks and trade names, see MPEP § 608.01(v).

608.01(q) Substitute or Rewritten Specification [R-2]

37 CFR 1.125. Substitute specification.

(a) If the number or nature of the amendments or the legibility of the application papers renders it difficult to consider the application, or to arrange the papers for printing or copying, the Office may require the entire specification, including the claims, or any part thereof, be rewritten.

**>

(b) Subject to § 1.312, a substitute specification, excluding the claims, may be filed at any point up to payment of the issue fee if it is accompanied by a statement that the substitute specification includes no new matter.

(c) A substitute specification submitted under this section must be submitted with markings showing all the changes relative to the immediate prior version of the specification of record. The text of any added subject matter must be shown by underlining the added text. The text of any deleted matter must be shown by strike-through except that double brackets placed before and after the deleted characters may be used to show deletion of five or fewer consecutive characters. The text of any deleted subject matter must be shown by being placed within double brackets if strike-through cannot be easily perceived. An accompanying clean version (without markings) must also be supplied. Numbering the paragraphs of the specification of record is not considered a change that must be shown pursuant to this paragraph.<

(d) A substitute specification under this section is not permitted in a reissue application or in a reexamination proceeding.

The specification is sometimes in such faulty English that a new specification is necessary; in such instances, a new specification should be required.

Form paragraph 6.28 may be used where the specification is in faulty English.

¶ 6.28 Idiomatic English

A substitute specification in proper idiomatic English and in compliance with 37 CFR 1.52(a) and (b) is required. The substitute specification filed must be accompanied by a statement that it contains no new matter.

37 CFR 1.125(a) applies to a substitute specification required by the Office. If the number or nature of the amendments or the legibility of the application papers renders it difficult to consider the application, or to arrange the papers for printing or copying, the Office may require the entire specification, including the claims, or any part thereof be rewritten.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of: Marcos C. Tzannes)	Group Art Unit: 2611
Application No.: 12/769,193)	Examiner: CORRIELUS, Jean B.
Filed: April 28, 2010)	Confirmation No.: 7092
Atty. File No.: 5550-52-PUS-CON)	

For: IMPULSE NOISE MANAGEMENT

AMENDMENT AFTER FINAL

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants submit this Amendment After Final to address the Final Office Action having a mailing date of May 25, 2011 and further to address the Notice of Appeal filed November 23, 2011. Please credit any overpayment or charge any underpayment to Deposit Account No. 19-1970.

Please amend the above-identified patent application as follows:

Amendments to the Claims are shown in the listing of claims which begins on page 2 of this paper.

Remarks begin on page 4 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a transceiver, a method of adapting an impulse noise protection capability of the transceiver during steady-state communication or initialization comprising:

transmitting or receiving, by the transceiver, using a first forward error correction setting;

transmitting or receiving a flag signal; and

switching to transmitting or receiving using a second forward error correction setting based at least on the flag signal,

wherein:

the first forward error correction setting comprises at least one first forward error correction parameter value,

the ~~first~~second forward error correction setting comprises at least one second forward error correction parameter value, different than the first forward error correction parameter value, and

the switching occurs ~~on a~~ on a pre-defined forward error correction codeword boundary following the flag signal.

2.-72. (Cancelled)

73. (Currently Amended) A system capable of adapting an impulse noise protection capability of a transceiver during steady-state communication or initialization comprising:

a transceiver that is capable of: transmitting or receiving using a first forward error correction setting,

transmitting or receiving a flag signal, and

switching to transmitting or receiving using a second forward error correction setting based at least on the flag signal,

wherein:

the first forward error correction setting comprises at least one first forward error correction parameter value,

the ~~first~~second forward error correction setting comprises at least one second forward error correction parameter value, different than the first forward error correction parameter value, and

the switching occurs ~~on~~a on a pre-defined forward error correction codeword boundary following the flag signal.

74. (Currently Amended) A non-transitory computer-readable information storage media having stored thereon instructions, that if executed by a processor, cause to be performed a method for adapting an impulse noise protection capability of a transceiver during steady-state communication or initialization comprising:

transmitting or receiving, by the transceiver, using a first forward error correction setting;

transmitting or receiving a flag signal; and

switching to transmitting or receiving using a second forward error correction setting based at least on the flag signal,

wherein:

the first forward error correction setting comprises at least one first forward error correction parameter value,

the ~~second~~first forward error correction setting comprises at least one second forward error correction parameter value, different than the first forward error correction parameter value, and

the switching occurs ~~on~~a on a pre-defined forward error correction codeword boundary following the flag signal.

REMARKS

Applicant respectfully requests reconsideration of this application as amended.

The Examiner is thanked for the thorough review of the subject application.

By this Amendment, the objection to the claims has been address in accordance with the Examiner's interpretation.

Regarding the objection to the claimed term "capable of," Applicant respectfully submits with reference to the United States Court of Appeals for the Federal Circuit decision in ENZO BIOCHEM, INC., ENZO LIFE SCIENCES, INC., and YALE UNIVERSITY, v. APPLERA CORP. and TROPIX, INC., the term "capable of" is definite.

Withdrawal of the objection of claims 1, 73 and 74 are respectfully requested.

Regarding the rejection of claims 1, 73 and 74 under 35 U.S.C. 112, second paragraph, the Examiner has misinterpreted the claims.

As is clear at least from Figs. 5-6 and the corresponding portions of the specification, the claims are clear and do distinctly claim features of the invention. The office Action states: "the switching to "receiving" would not be based on the "flag signal" since the transceiver is not disclosed to receive its own signal (i.e. the flag signal)."

Applicant respectfully submits this conclusion is incorrect.

There is nothing in the claim that states the transceiver is receiving its own flag signal. Rather, as discussed in accordance with one exemplary embodiment shown in Fig. 6 an exemplary method of synchronization using a flag signal is disclosed. In particular, control begins in step S600 and continues to step S610. In step S610, the modems enter Showtime using the first FIP parameters. Next, in step S620, a message is exchanged indicating the new FIP settings. Then, in step S630, the transmitter forwards to the receiver a flag signal indicating when the new FIP settings are to be used. At step S640, and at a predefined change time following the transmission of the flag signal, the transmitter begins transmission using the new FIP parameters. Next, at step S650, at the predefined change time following the reception of the flag signal, the receiver commences reception utilizing the new FIP parameters. Control then continues to step S660 where Showtime communication continues with the control sequence ending at step S670.

Therefore, Applicant respectfully submits the claims, regardless of which alternative is chosen, are not only fully supported by the specification but would be clear and definite to one of ordinary skill in the art.

Withdrawal of the rejection is respectfully requested.

Regarding the art-based rejection of claim 73 in view of Cioffi, Cioffi fails to teach, suggest or disclose each and every feature of amended claim 73. More specifically, Cioffi is directed towards adaptive FEC coding which is used to adjust the codeword composition of FEC codewords in a communication system. A codeword composition ratio may be adjusted in response to variance of a measured transmission error value from a target transmission error value in the system. The codeword composition ratio may be any quantity or value that represents the relation between the payload and parity bytes in the applicable FEC coding scheme. Adjustment of the codeword composition ratio may be adjusting parameters such as the N, K and/or R values in ADSL1 systems or the INP and/or maximum interleaving delay values in ADSL2 systems. A controller may be used to monitor, analyze and adjust the various values used in adaptively managing FEC coding.

Cioffi at least fails to teach the following:

- a flag signal,
- switching to using a second error correction setting based at least on the flag signal,
- a first forward error correction setting comprising at least one first forward error correction parameter value,
- a first forward error correction setting comprising at least one second forward error correction parameter value, different than the first forward error correction parameter value, and
- that the switching occurs on a on a pre-defined forward error correction codeword boundary following the flag signal.

Even more specifically, since Cioffi fails to make any mention of “flag signal,” or “codeword boundary,” Cioffi can not anticipate or render obvious the feature of switching occurring on a pre-defined forward error correction codeword boundary following the flag signal.

While it is assumed claims 1 and 74 are allowable, comparable arguments can be made for them as well in terms of how they are distinguishable from Cioffi.

With all objections and rejections having been overcome, Applicant respectfully submits the application is in condition for allowance. A prompt notice of allowance is respectfully solicited.

Should the Examiner believe anything further is desirable in order to place the application in even better condition for allowance, the Examiner is encouraged to contact Applicants undersigned representative at the telephone number listed below.

The Commissioner is hereby authorized to charge to deposit account number 19-1970 any fees under 37 CFR § 1.16 and 1.17 that may be required by this paper and to credit any overpayment to that Account. If any extension of time is required in connection with the filing of this paper and has not been separately requested, such extension is hereby petitioned.

Respectfully submitted,

SHERIDAN ROSS P.C.

Date: 6 Feb 18

By: 

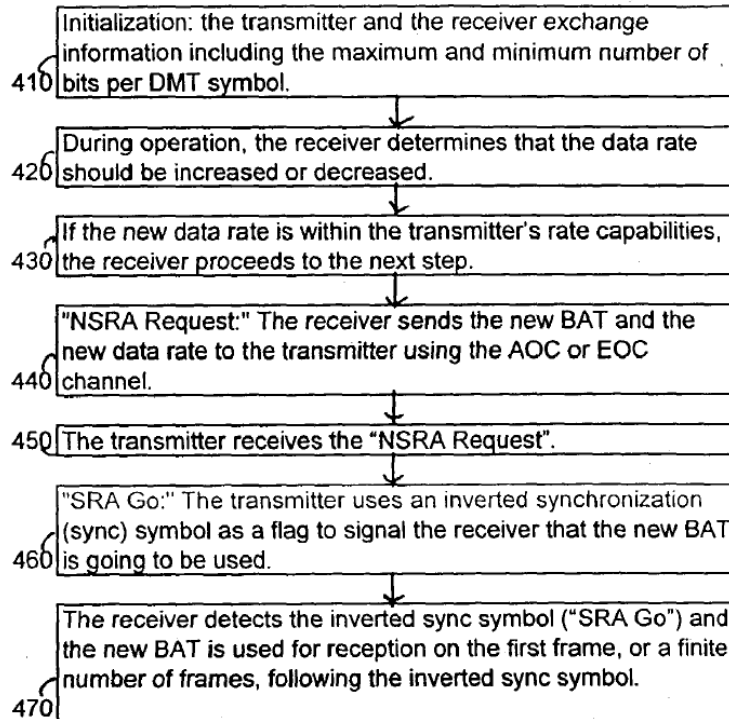
Jason H. Vick
Reg. No. 45,285
1560 Broadway, Suite 1200
Denver, Colorado 80202
Telephone: 303-863-9700



US 20030112884A1

(19) **United States**(12) **Patent Application Publication**
Tzannes(10) **Pub. No.: US 2003/0112884 A1**(43) **Pub. Date: Jun. 19, 2003**(54) **METHOD FOR SEAMLESSLY CHANGING
POWER MODES IN AN ADSL SYSTEM**filed on Oct. 22, 1999. Provisional application No.
60/177,081, filed on Jan. 19, 2000.(75) Inventor: **Marcos Tzannes, Orinda, CA (US)****Publication Classification**Correspondence Address:
NIXON PEABODY, LLP
8180 GREENSBORO DRIVE
SUITE 800
MCLEAN, VA 22102 (US)(51) **Int. Cl.⁷** **H04L 27/28; H04B 1/38**
(52) **U.S. Cl.** **375/260; 375/219**(57) **ABSTRACT**(73) Assignee: **Aware, Inc., Bedford, ME (US)**(21) Appl. No.: **10/351,402**(22) Filed: **Jan. 27, 2003****Related U.S. Application Data**(62) Division of application No. 09/522,870, filed on Mar.
10, 2000.(60) Provisional application No. 60/124,222, filed on Mar.
12, 1999. Provisional application No. 60/161,115,A DMT system and method with the capability to adapt the
system bit rate on-line in a seamless manner. The DMT
system provides a robust and fast protocol for completing
this seamless rate adaptation. The DMT system also pro-
vides a framing and encoding method with reduced over-
head compared to conventional DMT systems. The DMT
system and method provide seamless rate adaptation with
the provision of different power levels. This framing and
encoding method enables a system with seamless rate adap-
tation capability. The system and method of the invention
can be implemented in hardware, or alternatively in a
combination of hardware and software.**Receiver Initiated NSRA**

400



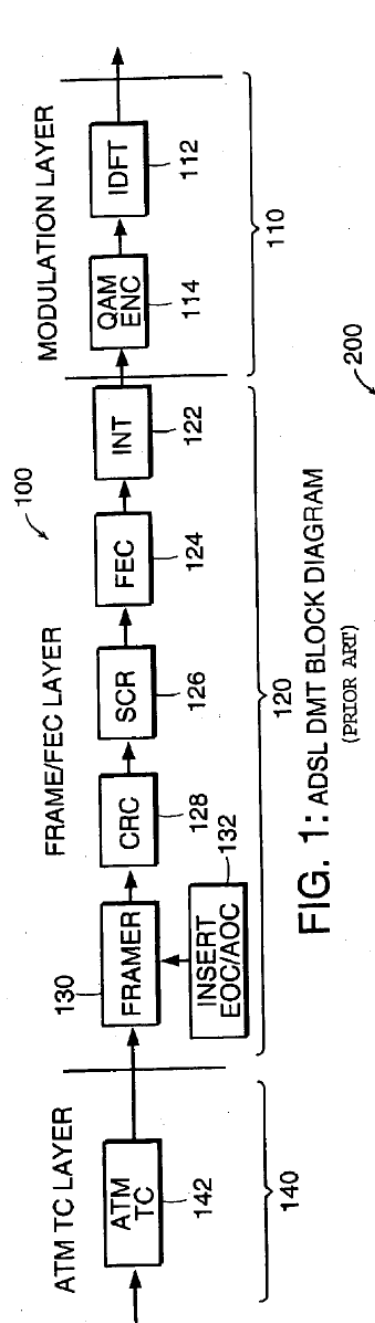


FIG. 1: ADSL DMT BLOCK DIAGRAM

(PRIOR ART)

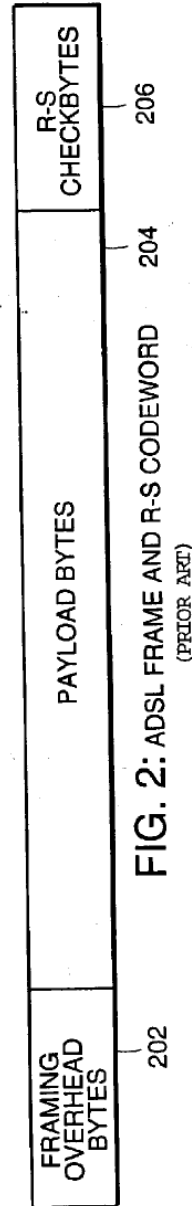


FIG. 2: ADSL FRAME AND R-S CODEWORD

(PRIOR ART)

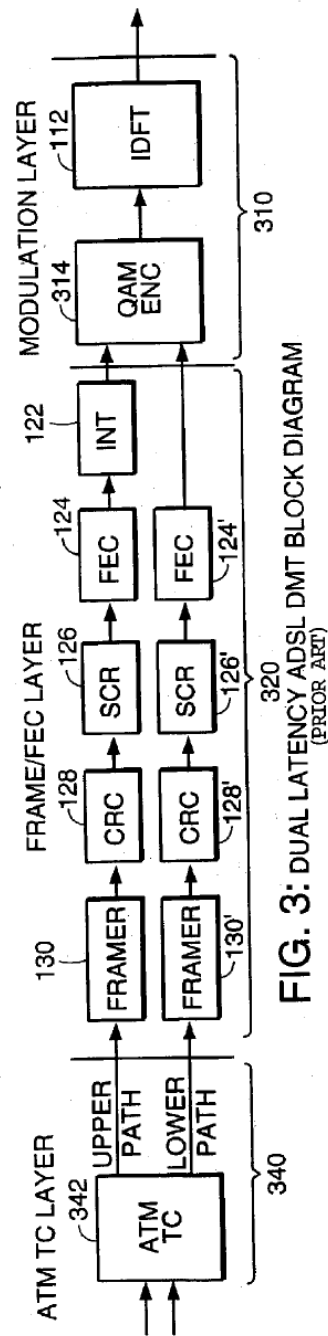


FIG. 3: DUAL LATENCY ADSL DMT BLOCK DIAGRAM

(PRIOR ART)

FIG. 4

Receiver Initiated NSRA

400

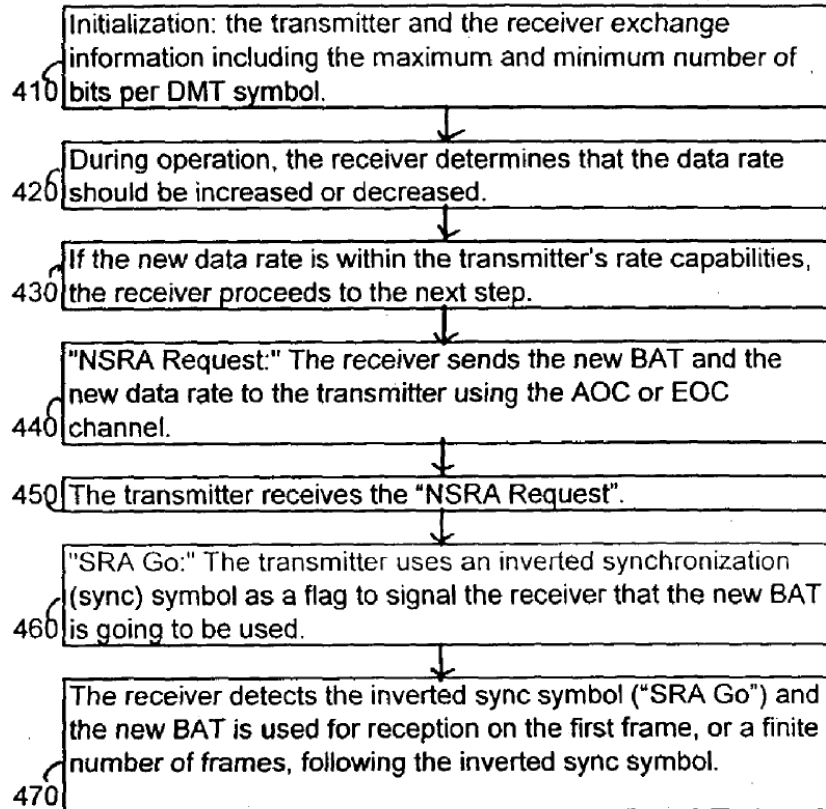


FIG. 5

Transmitter Initiated NSRA

500

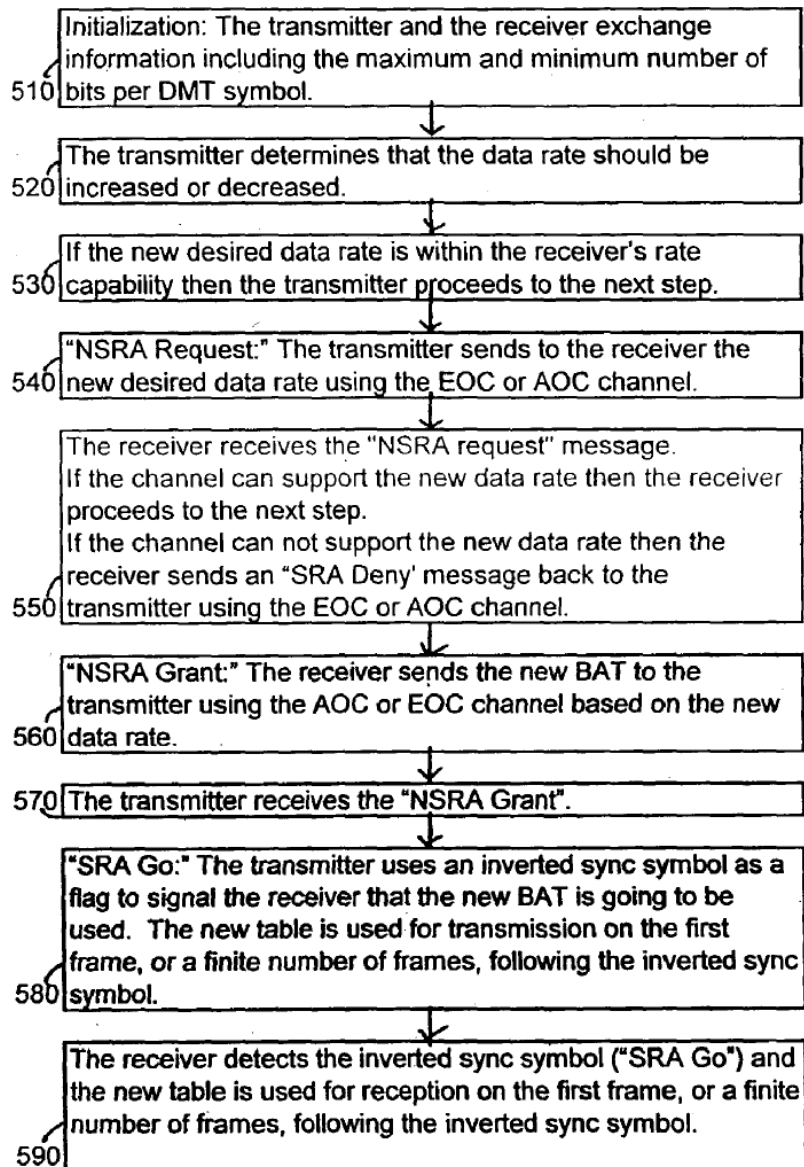


FIG. 6

Receiver Initiated FSRA

600

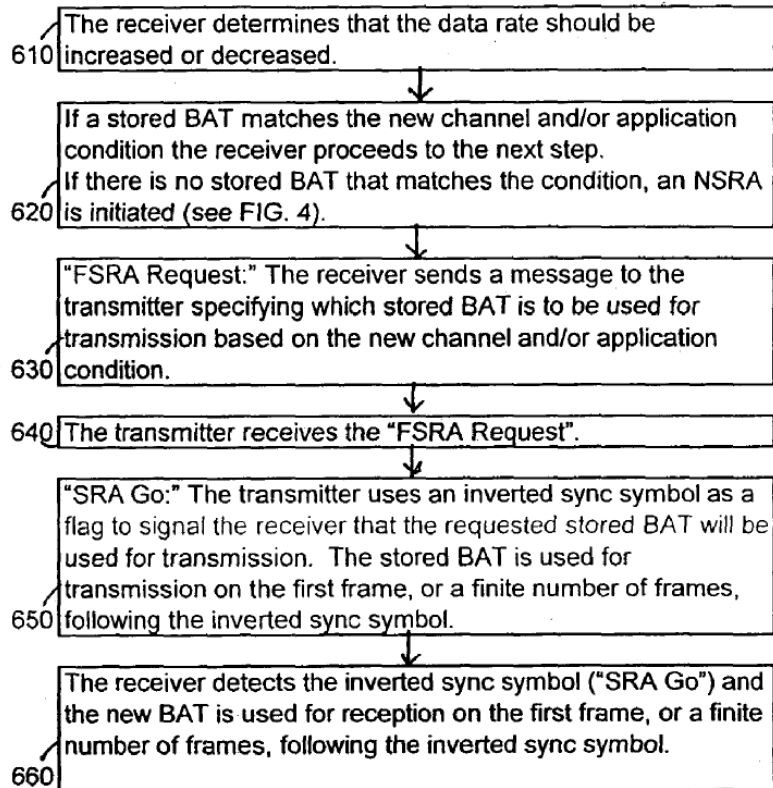
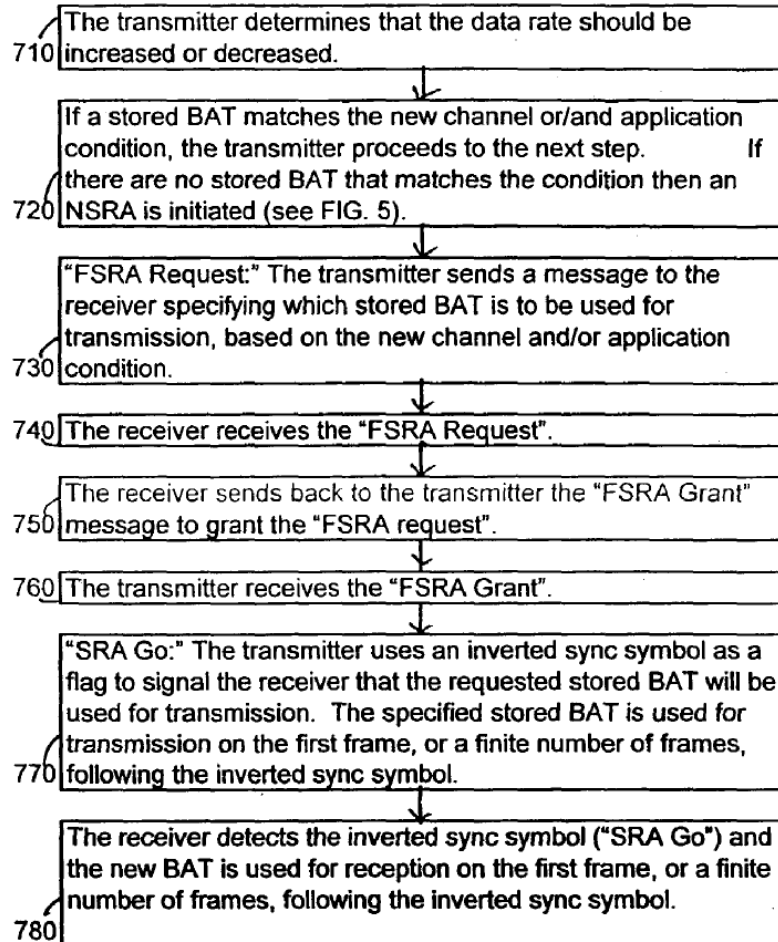


FIG. 7

Transmitter Initiated FSRA

700



US 2003/0112884 A1

Jun. 19, 2003

1

METHOD FOR SEAMLESSLY CHANGING POWER MODES IN AN ADSL SYSTEM**RELATED APPLICATIONS**

[0001] This application claims the benefit of and priority to U.S. provisional application Serial No. 60/124,222, filed Mar. 12, 1999, entitled "Seamless Rate Adaptive (SRA) ADSL System", U.S. provisional application Serial No. 60/161,115, filed Oct. 22, 1999, entitled "Multicarrier System with Stored Application Profiles", and U.S. provisional application Serial No. 60/171,081, filed Jan. 19, 2000, entitled "Seamless Rate Adaptive (SRA) Multicarrier Modulation System and Protocols," which copending provisional applications are incorporated herein by reference in their entirety.

FIELD OF THE INVENTION

[0002] This invention relates generally to communication systems and methods using multicarrier modulation. More particularly, the invention relates to communication multicarrier systems and methods using rate adaptive multicarrier modulation.

BACKGROUND OF THE INVENTION

[0003] Multicarrier modulation (or Discrete Multitone Modulation (DMT)) is a transmission method that is being widely used for communication over difficult media. Multicarrier modulation divides the transmission frequency band into multiple subchannels (carriers), with each carrier individually modulating a bit or a collection of bits. A transmitter modulates an input data stream containing information bits with one or more carriers and transmits the modulated information. A receiver demodulates all the carriers in order to recover the transmitted information bits as an output data stream.

[0004] Multicarrier modulation has many advantages over single carrier modulation. These advantages include, for example, a higher immunity to impulse noise, a lower complexity equalization requirement in the presence of multipath, a higher immunity to narrow band interference, a higher data rate and bandwidth flexibility. Multicarrier modulation is being used in many applications to obtain these advantages, as well as for other reasons. Applications include Asymmetric Digital Subscriber Line (ADSL) systems, Wireless LAN systems, Power Line communications systems, and other applications. ITU standards G.992.1 and G.992.2 and the ANSI T1.413 standard specify standard implementations for ADSL transceivers that use multicarrier modulation.

[0005] The block diagram 100 for a standard compliant ADSL DMT transmitter known in the art is shown in FIG. 1. FIG. 1 shows three layers: the Modulation layer 110, the Framer/FEC layer 120, and the ATM TC layer 140, which are described below.

[0006] The Modulation layer 110 provides functionality associated with DMT modulation. DMT modulation is implemented using an Inverse Discrete Fourier Transform (IDFT) 112. The IDFT 112 modulates bits from the Quadrature Amplitude Modulation (QAM) 114 encoder into the multicarrier subchannels. ADSL multicarrier transceivers modulate a number of bits on each subchannel, the number

of bits depending on the Signal to Noise Ratio (SNR) of that subchannel and the Bit Error Rate (BER) requirement of the link. For example, if the required BER is 1×10^{-7} (i.e., one bit in ten million is received in error on average) and the SNR of a particular subchannel is 21.5 dB, then that subchannel can modulate 4 bits, since 21.5 dB is the required SNR to transmit 4 QAM bits with a 1×10^{-7} BER. Other subchannels can have a different SNR and therefore may have a different number of bits allocated to them at the same BER. The ITU and ANSI ADSL standards allow up to 15 bits to be modulated on one carrier.

[0007] A table that specifies how many bits are allocated to each subchannel for modulation in one DMT symbol is called a Bit Allocation Table (BAT). A DMT symbol is the collection of analog samples generated at the output of the IDFT by modulating the carriers with bits according to the BAT. The BAT is the main parameter used in the Modulation layer 110 of FIG. 1. The BAT is used by the QAM 114 and IDFT 112 blocks for encoding and modulation. Table 1 shows an example of a BAT for a DMT system with 16 subchannels.

TABLE 1

Example of BAT for multicarrier system with 16 subchannels

Subchannel Number	Bits per Subchannel
1	5
2	9
3	3
4	2
5	4
6	0
7	5
8	7
9	8
10	3
11	0
12	5
13	6
14	8
15	4
16	3
Total bits PerDMT symbol	80

[0008] In ADSL systems the DMT symbol rate is approximately 4 kHz. This means that a new DMT symbol modulating a new set of bits, using the modulation BAT, is transmitted every 250 microseconds. If the BAT in table 1, specifies 80 bits modulated in one DMT symbol, were used at a 4 kHz DMT symbol rate the bit rate of the system would be $4000 \times 80 = 320$ kilobits per second (kbps). The BAT determines the data rate of the system and is dependent on the transmission channel characteristics, i.e. the SNR of each subchannel in the multicarrier system. A channel with low noise (high SNR on each subchannel) will have many bits modulated on each DMT carrier and will thus have a high bit rate. If the channel conditions are poor, the SNR will be low and the bits modulated on each carrier will be few, resulting in a low system bit rate. As can be seen in Table 1, some subchannels may actually modulate zero bits. An example is the case when a narrow band interferer (such as AM broad-

US 2003/0112884 A1

Jun. 19, 2003

2

cast radio) is present at a subchannel's frequency and causes the SNR in that subchannel to be too low to carry any information bits.

[0009] The ATM TC layer 140 includes an Asynchronous Transfer Mode Transmission Convergence (ATM TC) block 142 that transforms bits and bytes in cells into frames.

[0010] The next layer in an ADSL system is the Frame/FEC layer 120, which provides functionality associated with preparing a stream of bits for modulation, as shown in FIG. 1. This layer contains the Interleaving (INT) block 122, the Forward Error Correction (FEC) block 124, the scrambler (SCR) block 126, the Cyclic Redundancy Check (CRC) block 128 and the ADSL Framer block 130. Interleaving and FEC coding provide impulse noise immunity and a coding gain. The FEC 124 in the standard ADSL system is a Reed-Solomon (R-S) code. The scrambler 126 is used to randomize the data bits. The CRC 128 is used to provide error detection at the receiver. The ADSL Framer 130 frames the received bits from the ATM framer 142. The ADSL framer 130 also inserts and extracts overhead bits from module 132 for modem to modem overhead communication channels (known as EOC and AOC channels in the ADSL standards).

[0011] The key parameters in the Framer/FEC layer 120 are the size of the R-S codeword, the size (depth) of the interleaver (measured in number of R-S codewords) and the size of the ADSL frame. As examples, a typical size for an R-S codeword may be 216 bytes, a typical size for interleaver depth may be 64 codewords, and the typical size of the ADSL frame may be 200 bytes. It is also possible to have an interleaving depth equal to one, which is equivalent to no interleaving. In order to recover the digital signal that was originally prepared for transmission using a transmitter as discussed above, it is necessary to deinterleave the codewords by using a deinterleaver that performs the inverse process to that of the interleaver, with the same depth parameter. In the current ADSL standards there is a specific relationship between all of these parameters in a DMT system. Specifically, the BAT size, N_{BAT} (total number of bits in a DMT symbol) is fixed to be an integer divisor of the R-S codeword size, N_{FEC} , as expressed in equation (1):

$$N_{FEC} = S \times N_{BAT}, \text{ where } S \text{ is a positive integer greater than 0.} \quad (1)$$

[0012] This constraint can also be expressed as: One R-S codeword contains an integer number of DMT symbols. The R-S codeword contains data bytes and parity (checkbytes). The checkbytes are overhead bytes that are added by the R-S encoder and are used by the R-S decoder to detect and correct bit errors. There are R checkbytes in a R-S codeword. Typically, the number of checkbytes is a small percentage of the overall codeword size, e.g., 8%. Most channel coding methods are characterized by their coding gain, which is defined as the system performance improvement (in dB) provided by the code when compared to an uncoded system. The coding gain, of the R-S codeword depends on the number of checkbytes and the R-S codeword size. A large R-S codeword (greater than 200 bytes in a DMT ADSL system) along with a 16 checkbytes (8% of 200 bytes) will provide close to the maximum coding gain of 4 dB. If the codeword size is smaller and/or the percentage of checkbyte overhead is high (e.g. >30%) the coding gain may be very small or even negative. In general, it is best to have the

ADSL system operating with the largest possible R-S codeword (the maximum possible is 255 bytes) and approximately 8% redundancy.

[0013] There is also a specific relationship between the number of bytes in an ADSL frame, N_{FRAME} , and the R-S codeword size, N_{FEC} that is expressed in equation (2):

$$N_{FEC} = S \times N_{FRAME} + R; \text{ where } R \text{ is the number of R-S checkbytes in a codeword and } S \text{ is the same positive integer in Equation (1).} \quad (2)$$

[0014] It is apparent from equating the right-hand sides of equations (1) and (2) that the relationship expressed in equation (3) results:

$$N_{BAT} = N_{FRAME} + R/S. \quad (3)$$

[0015] The ADSL standard requires that the ratio (R/S) is an integer, i.e. there is an integer number of R-S checkbytes in every DMT-symbol (N_{BAT}). As described above, ADSL frames contain overhead bytes (not part of the payload) that are used for modem to modem communications. A byte in an ADSL frame that is used for the overhead channel cannot be used for the actual user data communication, and therefore the user data rate decreases accordingly. The information content and format of these channels is described in the ITU and ANSI standards. There are several framing modes defined in ADSL standards; Depending on the framing mode, there are more or fewer overhead bytes in one ADSL frame. For example, standard Framing Mode 3 has 1 overhead byte per ADSL frame.

[0016] Equations (1), (2) and (3) demonstrate that the parameter restrictions imposed by the standards result in the following conditions:

[0017] 1. All DMT symbols have a fixed number of overhead framing bytes that are added at the ADSL framer. For example, in framing mode #3 there is 1 overhead framing byte per DMT symbol.

[0018] 2. There is a minimum of 1 R-S checkbyte per DMT symbol.

[0019] 3. The maximum number of checkbytes according to ITU Standard G.992.2 (8) and ITU Standards G.992.2 and T1.413 (16) limits the maximum codeword size to $8 \times N_{BAT}$ for G.992.2, and to $16 \times N_{BAT}$ for G.992.1 and T1.413.

[0020] 4. An ADSL modem cannot change the number of bits in a DMT symbol (N_{BAT}) without making the appropriate changes to the number of bytes in a R-S codeword (N_{FEC}) and an ADSL frame (N_{FRAME}).

[0021] The above four restrictions cause performance limitations in current ADSL systems.

[0022] In particular, because of condition #1 every DMT symbol has a fixed number of overhead framing bytes. This is a problem when the data rate is low and the overhead framing bytes consume a large percentage of the possible throughput resulting in a lower payload. For example, if the data rate supported by the line is 6.144 Mbps, this will result in a DMT symbol with about 192 bytes per symbol ($192 \times 8 \times 4000 = 6144000$ bps). In this case, one overhead framing byte would consume $1/192$ or about 0.5% of the available throughput. But if the data rate is 128 kbps or 4

US 2003/0112884 A1

Jun. 19, 2003

3

bytes per symbol the overhead framing byte will consume $\frac{1}{4}$ or 25% of the available throughput. Clearly this is undesirable.

[0023] Condition #2 will cause the same problems as condition #1. In this case, the overhead framing byte is replaced by the R-S checkbyte.

[0024] Condition #3 will not allow the construction of large codewords when the data rate is low. R-S codewords in ADSL can have a maximum of 255 bytes. The maximum coding gain is achieved when the codeword size is near the maximum 255 bytes. When the data rate is low, e.g. 128 kbps or 4 bytes per symbol, the maximum codeword size will be $8 \times 4 = 32$ bytes for G.992.2 systems and $16 \times 4 = 64$ bytes for G.992.1 and T1.413 systems. In this case the coding gain will be substantially lower than for large codewords approaching 255 bytes.

[0025] In general, if the data rate is low, e.g. 128 kbps or 4 byte per symbol, the above conditions will result in 1 byte being used for overhead framing, and 1 byte being consumed by a R-S checkbyte. Therefore 50% of the available throughput will not be used for payload and the R-S codeword size will be at most 64 bytes, resulting in negligible coding gain.

[0026] Condition #4 effects the ability of the modem to adapt its transmission parameters on-line in a dynamic manner.

[0027] G.992.1 and T1.413 specify a mechanism to do on-line rate adaptation, called Dynamic Rate Adaptation (DRA), but it is clearly stated in these standards that the change in data rate will not be seamless. In general current ADSL DMT modems use Bit Swapping and dynamic rate adaptation (DRA) as methods for on-line adaptation to channel changes. Bit Swapping is specified in the ITU and ANSI standards as method for modifying the number of bits allocated to a particular. Bit Swapping is seamless, i.e., it does not result in an interruption in data transmission and reception. But, Bit Swapping does not allow the changing of data rates. Bit Swapping only allows the changing the number of bits allocated to carriers while maintaining the same data rate. This is equivalent to changing the entries in the BAT table without allowing the total number of bits (N_{BAT}) in the BAT to increase or decrease.

[0028] DRA enables a change in data rate, but is not seamless. DRA is also very slow because it requires the modem located in the Central Office (CO) to make the final decision on the data rate configuration. This model, with (the CO being the master), is common among ADSL modems that are designed to provide a service offered by the telephone company, and controlled by the telephone company.

[0029] Both Bit Swapping and DRA use a specific protocol that is specified in ANSI T1.413, G.992.1 and G.992.2 for negotiating the change. This protocol negotiates the parameters using messages that are sent via an AOC channel, which is an embedded channel. This protocol is sensitive to impulse noise and high noise levels. If the messages are corrupted the transmitter and receiver can enter a state where they are using different transmission parameters (e.g., BAT, data rate, R-S codeword length, interleaver depth, etc). When two communication modems enter a state of mismatched transmission parameters, data will be received in error and the modems will eventually be required to take

drastic measures (such as full reinitialization) in order to restore error free transmission. Drastic measures such as full reinitialization will result in the service being dropped for approximately 10 seconds, which is the time required for a standards compliant ADSL modem to complete a full initialization.

[0030] A transceiver has both a transmitter and a receiver. The receiver includes the receiver equivalent blocks of the transmitter shown in FIG. 1. The receiver has modules that include a decoder, a deinterleaver and a demodulator. In operation, the receiver accepts a signal in analog form that was transmitted by a transmitter, optionally amplifies the signal in an amplifier, filters the signal to remove noise components and to separate the signal from other frequencies, converts the analog signal to a digital signal through the use of an analog to digital converter, demodulates the signal to generate the received bits stream from the carrier sub-channels by the use of a demodulator, deinterleaves the bits stream by the use of a deinterleaver, performs the FEC decoding to correct errors in the bit stream by use of an FEC decoder, descrambles the bit stream by use of a descrambler, and detects bit errors in the bit stream by use of a CRC. Various semiconductor chip manufacturers supply hardware and software that can perform the functions of a transmitter or a receiver, or both.

[0031] It is therefore apparent that there is a need for an improved DMT transmission system. It is therefore a principle object of the invention to provide an improved DMT transmission system that overcomes the problems discussed above.

SUMMARY OF THE INVENTION

[0032] According to the principles of the invention, ADSL DMT systems and methods are provided that change transmission bit rates in a seamless manner during operation. The ADSL DMT systems and methods operate according to protocols that allow the seamless change of transmission bit rates during operation to be initiated by either the transmitter or the receiver. The ADSL DMT systems and methods provide for seamless changes of transmission bit rates during operation that change transmission bit rates between power levels that range from full power to low power.

[0033] In one aspect, the invention relates to a method for seamlessly entering a second power mode from a first power mode. The method uses a multicarrier transmission system that includes a transmitter and a receiver. The transmitter and receiver use a first bit allocation table to transmit a plurality of codewords at a first transmission bit rate in a first power mode. The plurality of codewords have a specified codeword size and include a specified number of parity bits for forward error correction, and a specified interleaving parameter for interleaving the plurality of codewords. The method involves storing a second bit allocation table at the receiver and at the transmitter for transmitting codewords at a second transmission in the second power mode. The method includes synchronizing use of the second bit allocation table between the transmitter and receiver, and entering the second power mode by using the second bit allocation table to transmit codewords. In order to achieve a seamless change in power mode, the specified interleaving parameter, the specified codeword size, and the specified number of parity bits for forward error correction used to

US 2003/0112884 A1

Jun. 19, 2003

4

transmit codewords in the first power mode are also used to transmit codewords in the second power mode.

[0034] In one embodiment, the synchronizing includes sending a flag signal. In another embodiment, the flag signal is a predefined signal. In a further embodiment, the predefined signal is a sync symbol with a predefined phase shift. In a still further embodiment, the predefined signal is an inverted sync symbol. In another embodiment, the transmitter transmits the flag signal to the receiver. In a different embodiment, the receiver transmits the flag signal to the transmitter. In another embodiment, the second power mode is a low power mode.

[0035] In another embodiment, the method further involves allocating zero bits to carrier signals to achieve a transmission bit rate of approximately zero kilobits per second in the low power mode. In another embodiment, the method further includes transmitting a pilot tone for timing recovery when operating in the low power mode. In still another embodiment, the method further comprises periodically transmitting a sync symbol when operating in the low power mode.

[0036] In still another embodiment, the method further includes using the first bit allocation table for transmitting a plurality of DMT symbols in the first power mode and switching to the second bit allocation table for transmitting the plurality of the DMT symbols in the second power mode. The second bit allocation table is used for transmission starting with a predetermined one of the DMT symbols that follows the transmission of the flag signal. In another embodiment, the predetermined DMT symbol is the first DMT symbol that follows the transmission of the flag signal.

[0037] In another embodiment, the second power mode is a full power mode. In still another embodiment, the first power mode is a full power mode, and the second power mode is a low power mode. In another embodiment, the first power mode is a low power mode, and the second power mode is a full power mode.

BRIEF DESCRIPTION OF THE DRAWINGS

[0038] The objects and features of the invention can be better understood with reference to the drawings described below. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention.

[0039] FIG. 1 is a block diagram for a standard compliant ADSL DMT transmitter known in the prior art.

[0040] FIG. 2 is an exemplary embodiment of an ADSL frame and R-S codeword.

[0041] FIG. 3 is a block diagram for a dual latency ADSL DMT transmitter.

[0042] FIG. 4 is a flow chart that depicts an embodiment of a process in which a Normal Seamless Rate Adaptive (NSRA) transmission bit rate change is initiated by a receiver according to the principles of the invention.

[0043] FIG. 5 is a flow chart that depicts an embodiment of a process in which a Normal Seamless Rate Adaptive (NSRA) transmission bit rate change is initiated by a transmitter according to the principles of the invention.

[0044] FIG. 6 is a flow chart that depicts an embodiment of a process in which a Fast Seamless Rate Adaptive (FSRA) transmission bit rate change is initiated by a receiver according to the principles of the invention.

[0045] FIG. 7 is a flow chart that depicts an embodiment of a process in which a Fast Seamless Rate Adaptive (FSRA) transmission bit rate change is initiated by a transmitter according to the principles of the invention.

DETAILED DESCRIPTION

[0046] The principles of the invention may be employed using transceivers that include a transmitter, such as that described in FIG. 1 above, and a receiver. In general terms, an ADSL system includes both a transmitter and a receiver for each communication in a particular direction. In the discussion that follows, an ADSL DMT transmitter accepts digital input and transmits analog output over a transmission line, which can be a twisted wire pair, for example. The transmission can also occur over a medium that includes other kinds of wires, fiber optic cable, and/or wireless connections. In order to utilize the transmitted signal, a second transceiver at the remote end of the transmission line includes a receiver that converts the received analog signal into a digital data stream for use by devices such as computers or digital televisions, for example. For bidirectional communication using a pair of transceivers, each transceiver includes a transmitter that sends information to the receiver of the other member of the pair, and a receiver that accepts information transmitted by the transmitter of the other member of the pair.

[0047] This invention describes a DMT system with the capability to adapt the system bit rate on-line in a seamless manner. The DMT system also provides a robust and fast protocol for completing this seamless rate adaptation. The DMT system also provides a framing and encoding method with reduced overhead compared to conventional DMT systems. This new framing and encoding method also enables a system with seamless rate adaptation capability.

[0048] It may be desirable to change the modem data rate after training due to a change in the channel characteristics or because the application running over ADSL has changed. Examples of changing channel characteristics include changes in the noise on the line, changes in the crosstalk from other services in the bundle or on the same line, changes in the levels and presence of Radio Frequency Interference ingress, changes in the line impedance due to temperature changes, changes in the state of equipment on the line (e.g. a phone going from on-hook to off hook, or vice versa), and the like. Examples of changes in applications include power down modes for a PC, a user changing from Internet browsing to two-way video conferencing, a user changing from internet browsing to voice over DSL with or without internet browsing, and the like. It is often desirable or required to change the data rate of the modem. It is highly desirable that this data rate change occurs in a "seamless" manner, i.e., without data bit errors or an interruption in service. However DMT ADSL modems specified in the prior art standards are not capable of performing seamless data rate adaptation.

[0049] Condition #4 described previously, does not allow the size of the BAT to change without modifying the R-S coding, interleaving and framing parameters. If the BAT, and

US 2003/0112884 A1

Jun. 19, 2003

5

N_{BAT} could be modified during operation, i.e., if more or fewer bits were allocated to carriers in a DMT symbol, the data rate could be changed. Condition #4 requires that when the number of bits N_{BAT} in the BAT changes the size of the R-S codeword (and therefore interleaving parameters) must also be modified. Modifying the interleaving and coding parameters on-line requires re-initializing the interleaver. Re-initialization of the interleaver always results in a “flushing” of the interleave memory. This flushing of memory will result in data errors and the transition will not be seamless.

[0050] In order to allow a DMT ADSL transmission system to change data rate seamlessly, the invention relates to the following:

- [0051] 1. a more efficient method for framing and encoding the data, that results in less overhead data bits per DMT symbol, thereby increasing the user bit rate;
- [0052] 2. a new ADSL system with the ability to dynamically adapt the data rate on-line (e.g., during operation) in a seamless manner; and
- [0053] 3. a new robust and fast protocol for completing such a seamless rate adaptation, so a data rate change can occur successfully even in the presence of high levels of noise.

[0054] Constant Percentage Overhead Framing

[0055] In one embodiment, a framing method is described that decreases the overhead (non-payload) data in DMT ADSL systems. FIG. 2 shows a diagram 200 representative of an ADSL frame and R-S codeword that includes at least one framing overhead byte 202, one or more payload bytes 204, and one or more checkbytes 206. This framing method also enables seamless rate adaptation. As described above current ADSL systems place restrictions and requirements on the ADSL frames, R-S codewords and DMT symbols. A system implemented according to the principles of the invention de-couples ADSL frames and R-S codewords from DMT symbols. This decoupling results in a system that has lower overhead data per DMT symbol and also can complete online rate adaptations in a seamless manner. According to the principles of the invention, ADSL frames and R-S codewords are constructed to have the same length and to be aligned (see FIG. 2). The R-S codeword is made sufficiently large enough to maximize the coding gain. The size of the R-S codeword (and therefore ADSL frame) can be negotiated at startup or fixed in advance. A fixed number of R-S checkbytes and overhead framing bytes are included in an ADSL frame. These parameters can also be negotiated at startup or fixed in advance.

[0056] Unlike DMT symbols of the prior art, DMT symbols produced in accordance with the principles of the invention are not aligned with ADSL frames and R-S codewords. Also the number of bits in a DMT symbol depends solely on the data rate requirements and configurations and is de-coupled from the R-S codeword size, the interleaver depth and the ADSL frame size. The number of bits in a DMT symbol dictates the data rate of the modem independently of the other framing, coding or interleaving restrictions. Since overhead bytes are added at the ADSL frame layer, a DMT symbol does not necessarily contain a fixed number of overhead bytes. As the data rate gets lower, for example 128 kbps, the overhead data remains low. In

particular, this framing method assigns a fixed percentage of overhead data to the data stream, rather than a fixed number of overhead bytes. This percentage does not change when the data rate of the modem changes, (as is the case with current ADSL modems). Consider the following examples of conventional standard compliant framing methods.

Prior Art Example #1

[0057] The line capacity is 192 bytes per DMT symbol (6.144 Mbps). The codeword size is 192, which includes 16 checkbytes and 1 overhead framing byte, (assuming ANSI T1.413 framing mode #3). The total framing overhead (i.e., checkbytes+overhead framing bytes) per DMT symbol is $16+1=17$, and therefore the framing overhead is $17/192=8.8\%$ of the available throughput. In this case the framing overhead is reasonable.

Prior Art Example #2

[0058] The line capacity is 4 bytes (128 kbps). The codeword is constructed from 16 DMT symbols and is $16*4=64$ bytes. There are 16 R-S checkbytes (1 checkbyte per DMT symbol) and there is 1 overhead framing byte (assuming ANSI T1.413 framing mode #3). The total framing overhead (checkbytes+overhead framing bytes) per DMT symbol is $1+1=2$ bytes, and therefore the framing overhead is $2/4=50\%$ of the available throughput. This is highly inefficient.

[0059] Examples of embodiments of the framing method of the invention provide the following results, called the Constant Percentage Overhead Method:

Example #1

[0060] This is exactly the same as the standard compliant training example (Prior Art Example #1) given above. Codeword sizes, DMT symbol sizes and overhead are the same. Therefore the framing overhead is $17/192=8.8\%$ of the available throughput as well.

Example #2

[0061] The line capacity is 4 bytes (128 kbps). The codeword is constructed independently of the DMT symbol and therefore could be set to 192 bytes, (as an example). This is also the size of the ADSL frame. We use 16 R-S bytes and 1 overhead framing byte per codeword or ADSL frame. There are $192/4=48$ DMT symbols in 1 codeword. The total overhead (checkbytes+overhead framing bytes) per 48 DMT symbols is $1+16=17$ bytes or $17/48=0.35$ bytes per 1 DMT symbol. The framing overhead is $0.35/4=8.8\%$ of the available throughput.

[0062] From examples #1 and #2, it is apparent that the principles of the invention provide a method to achieve a framing overhead that is a constant percentage of the available throughput regardless of the data rate or the line capacity. In these examples, the framing overhead was 8.8% for both 6 Mbps and 128 kbps.

[0063] Seamless Rate Adaptation (SRA) System

[0064] Another benefit of the framing method described in this invention is that it enables seamless on-line rate adaptation. Seamless Rate Adaptation (SRA) is accomplished by changing the DMT symbol BAT, i.e. the number of bits allocated to each subchannel in the multicarrier system. As shown above modifying the BAT changes the number of bits

US 2003/0112884 A1

Jun. 19, 2003

6

per DMT symbol and results in a change in the data bit rate of the system. In one embodiment, the DMT symbol size is changed without modifying any of the RS coding, interleaving and framing parameters. This is possible because the Constant Percentage Overhead framing method described above removes the restrictions imposed by the prior art on the relation between DMT symbols and R-S codewords or ADSL frames. Since the R-S coding and interleaving parameters do not change, interleaver flushing and other problems associated with changing the parameters associated with these functions do not occur. The transceiver can adapt the data rate without errors or service interruption. The only parameter that needs to be adapted is the BAT.

[0065] The BAT needs to be changed at the transmitter and the receiver at exactly the same time, i.e., on exactly the same DMT symbol. If the transmitter starts using the new BAT for transmission before the receiver does, the data is not demodulated correctly and bit errors occur. Also, if the receiver changes to a new BAT before the transmitter does, the same errors can occur. For this reason the transition to the use of the new BAT for transmission and reception needs to be synchronized at the transmitter and the receiver. In one embodiment, the principles of the invention provide a protocol that enables the synchronized transition to the use of the new BAT.

[0066] It is also very important that this protocol is very robust in the presence of channel noise. For example, if the protocol fails and the receiver does not switch to the new BAT at the same time as the transmitter, then bit errors occur and the transition is not seamless. Furthermore, if the transmitter and receiver are using different BATs, it is very difficult to re-establish an error free link without performing a re-initialization of the connection, which results in an interruption in service of up to 10 seconds.

[0067] It is also important that the transition between BATs occurs very quickly, because the need to operate at a new data rate is usually instantaneous. As an example, at a constant data rate a sudden decrease in the channel SNR will increase the number of bits received in error. A change in data rate is required because of the reception of many bits in error. In this situation, it is desirable to change the data rate as soon as possible to get out of the state of receiving bits in error. As another example, a change in the applications being transported over the ADSL link can require a change in the data rate. For example if one user is browsing the Internet and then another user wishes to make a voice call over the flow of data bits using the Voice over DSL capability of the ADSL connection, it is necessary to quickly change the data rate of the system to accommodate the telephone call in addition to the existing traffic.

[0068] It is apparent from these requirements that it is necessary for the SRA protocol to provide:

- [0069] a. a method for synchronizing the transmitter and receiver transition to the new BAT;
- [0070] b. robust transition to the new data rate; and
- [0071] c. fast transition to the new data rate.

[0072] The principles of the invention provide two protocols that satisfy these requirements for seamless rate adaptation. The protocols are called the Normal SRA protocol and the Fast SRA protocol.

[0073] Normal SRA (NSRA) Protocol

[0074] Either the transmitter or the receiver can initiate the Normal SRA (NSRA) protocol.

[0075] Receiver Initiated NSRA

[0076] The receiver initiated NSRA involves the following steps:

[0077] 1. During initialization the transmitter and the receiver exchange information describing their maximum and minimum data rate capabilities. This corresponds to the maximum and minimum number of bits per DMT symbol.

[0078] 2. During operation, the receiver determines that the data rate should be increased or decreased.

[0079] 3. If the new data rate is within the transmitter's rate capabilities, the receiver proceeds to step 4.

[0080] 4. The receiver sends the new BAT and the new data rate to the transmitter using the AOC or EOC channel. This corresponds to "NSRA Request" by the receiver.

[0081] 5. The transmitter receives the "NSRA Request".

[0082] 6. The transmitter uses an inverted synchronization (sync) symbol as a flag to signal the receiver that the new BAT is going to be used. The new BAT is used for transmission on the first frame, or a finite number of frames, following the inverted sync symbol. The inverted sync symbol operates as a rate adaptation "SRA Go" message sent by the transmitter.

[0083] 7. The receiver detects the inverted sync symbol ("SRA Go") and the new BAT is used for reception on the first frame, or a finite number of frames, following the inverted sync symbol.

[0084] FIG. 4 shows a flow chart 400 depicting an embodiment of a process in which a Normal Seamless Rate Adaptive (NSRA) transmission bit rate change is initiated by a receiver according to the principles of the invention. In FIG. 4, the steps described in action boxes 410 through 470 correspond to the preceding discussion.

[0085] Transmitter Initiated NSRA

[0086] The transmitter initiated NSRA involves the following steps:

[0087] 1. During initialization the transmitter and the receiver exchange information describing their maximum and minimum capabilities regarding data rate. This corresponds to the maximum and minimum number of bits per DMT symbol.

[0088] 2. The transmitter determines that the data rate should be increased or decreased.

[0089] 3. If the new desired data rate is within the receiver's rate capability then the transmitter proceeds to step 4.

[0090] 4. The transmitter sends to the receiver the new desired data rate using the EOC or AOC channel. This is an "NSRA Request" message.

US 2003/0112884 A1

Jun. 19, 2003

7

[0091] 5. The receiver receives the NSRA request message. If the channel can support the new data rate then the receiver proceeds to step 6. If the channel can not support the new data rate then the receiver sends an "SRA Deny" message back to the transmitter using the EOC or AOC channel.

[0092] 6. The receiver sends the new BAT to the transmitter using the AOC or EOC channel based on the new data rate. This corresponds to an "NSRA Grant" request by the receiver.

[0093] 7. The transmitter receives the "NSRA Grant".

[0094] 8. The transmitter uses an inverted sync symbol as a flag to signal the receiver that the new BAT is going to be used. The new table is used for transmission on the first frame, or a finite number of frames, following the inverted sync symbol. The inverted sync signal operates as a rate adaptation "SRA Go" message sent by the transmitter.

[0095] 9. The receiver detects the inverted sync symbol ("SRA Go") and the new table is used for reception on the first frame, or a finite number of frames, following the inverted sync symbol.

[0096] FIG. 5 shows a flow chart 500 depicting an embodiment of a process in which a Normal Seamless Rate Adaptive (NSRA) transmission bit rate change is initiated by a transmitter according to the principles of the invention. In FIG. 5, the steps described in action boxes 510 through 590 correspond to the preceding discussion.

[0097] The rate adaptation only involves changing the number of bits in a DMT symbol by changing the BAT, and not the R-S codeword size, interleaver depth, or the ADSL frame size. This can be done without any interruption in data flow or introduction of data errors.

[0098] This protocol of the invention is faster than conventional rate adaptation methods because it does not require an extended handshake between the transmitter and the receiver in order to approve the new transmission parameters and rates. No extended handshake is needed because the data rate capabilities are known in advance and negotiated during startup. Also, the other parameters (such as R-S codeword length, interleaver depth, etc) are not changed during the data rate change using the new framing method.

[0099] This protocol of the invention is more robust than conventional rate adaptation techniques because it does not use the EOC or AOC channel to send the "SRA Go" message for synchronizing the transition to the new data rate. In conventional rate adaptation techniques, messages sent over the EOC and AOC channel can easily become corrupted by noise on the line. These overhead channels are multiplexed into the data stream at the framer and therefore are transmitted with quadrature amplitude modulation over a finite number of DMT subchannels. Impulse noise or other noise that occurs on the line can easily cause bit errors in the AOC channel message; the message can be lost. If the "SRA Go" message is corrupted and not received by the receiver, then the receiver does not know if the SRA request was granted or not. The transmitter, on the other hand, assumes the "SRA Go" message was received and switches to the new data rate

and transmission parameters. The receiver, which did not receive the grant message, does not know when to switch to the new rate. The modems are unsynchronized and data errors occur.

[0100] The protocol of the invention is robust because, unlike conventional rate adaptation techniques, the "SRA Go" message is not sent via an EOC or AOC message that can easily be corrupted. Instead the grant of the rate adaptation request is communicated via an inverted sync symbol. The sync symbol is defined in the ANSI and ITU standards as fixed non-data carrying DMT symbol that is transmitted every 69 symbols. The sync symbol is constructed by modulating all the DMT carriers with a predefined PN sequence using basic QPSK (2 bit QAM) modulation. This signal, which is used throughout the modem initialization process, has special autocorrelation properties that make possible the detection of the sync symbol and the inverted sync symbol even in highly noisy environments. An inverted sync symbol is a sync symbol in which the phase information in the QAM signal is shifted by 180 degrees. Other phase shifts (other than 180 degrees) of the sync symbol can be used as well for the SRA Go message. Using the sync symbol for the "SRA Go" message makes the rate adaptation protocol very robust even in the noisiest environments.

[0101] Fast SRA (FSRA) Protocol Using Stored BATs

[0102] The Fast SRA (FSRA) protocol seamlessly changes the data rate on the line faster than the NSRA protocol. This is important for certain applications that are activated and de-activated instantaneously over time or when sudden channel changes occur. In the FSRA protocol, "stored BATs" are used to speed up the SRA handshake and enable quick changes in data rate. Unlike profiles used in G.992.2, the stored BAT does not contain the R-S coding and interleaving parameters since these parameters are not effected when a rate change occurs using constant percentage overhead framing.

[0103] The BATs are exchanged using NSRA described in the previous section. After the one time NSRA is complete, and a BAT that is based on that particular channel condition or application condition is stored by both transceivers, the FSRA protocol can use the stored BAT to complete fast on-line rate adaptation. Stored BATs are labeled so that both the transmitter and receiver simply need to notify the other which table is to be used without actually having to transmit the information again. For example, the stored BAT may be numbered. The transmitter or receiver simply needs to tell the other transceiver which BAT table number is to be used for subsequent transmission. Either the receiver or the transmitter can initiate the FSRA protocol.

[0104] Receiver-Initiated FSRA

[0105] The receiver-initiated FSRA protocol involves the following steps:

[0106] 1. The receiver determines that the data rate should be increased or decreased.

[0107] 2. If a stored BAT matches the new channel and/or application condition the receiver proceeds to step 3. If there is no stored BAT that matches the condition, an NSRA is initiated (as described above).

[0108] 3. The receiver sends a message to the transmitter specifying which stored BAT is to be used for

US 2003/0112884 A1

8

Jun. 19, 2003

transmission based on the new channel and/or application condition. This corresponds to an "FSRA Request" by the receiver.

[0109] 4. The transmitter receives the "FSRA Request".

[0110] 5. The transmitter uses an inverted sync symbol as a flag to signal the receiver that the requested stored BAT will be used for transmission. The stored BAT is used for transmission on the first frame, or a finite number of frames, following the inverted sync symbol. The inverted sync signal corresponds to a rate adaptation "SRA Go" message sent by the transmitter.

[0111] 6. The receiver detects the inverted sync symbol ("SRA Go") and the new BAT is used for reception on the first frame, or a finite number of frames, following the inverted sync symbol.

[0112] FIG. 6 shows a flow chart 600 depicting an embodiment of a process in which a Fast Seamless Rate Adaptive (FSRA) transmission bit rate change is initiated by a receiver according to the principles of the invention. In FIG. 6, the steps described in action boxes 610 through 660 correspond to the preceding discussion.

[0113] Transmitter-initiated FSRA

[0114] The transmitter-initiated FSRA protocol involves the following steps:

[0115] 1. The transmitter determines that the data rate should be increased or decreased.

[0116] 2. If a stored BAT matches the new channel or/and application condition, the transmitter proceeds to step 3. If there are no stored BAT that matches the condition then an NSRA is initiated (as described above).

[0117] 3. The transmitter sends a message to the receiver specifying which stored BAT is to be used for transmission, based on the new channel and/or application condition. This corresponds to an "FSRA Request" by the transmitter.

[0118] 4. The receiver receives the "FSRA Request".

[0119] 5. The receiver sends back to the transmitter the "FSRA Grant" message to grant the "FSRA request".

[0120] 6. The transmitter receives the "FSRA Grant".

[0121] 7. The transmitter uses an inverted sync symbol as a flag to signal the receiver that the requested stored BAT will be used for transmission. The specified stored BAT is used for transmission on the first frame, or a finite number of frames, following the inverted sync symbol. The inverted sync signal corresponds to a rate adaptation "SRA Go" message sent by the transmitter.

[0122] 8. The receiver detects the inverted sync symbol ("SRA Go") and the new BAT is used for reception on the first frame, or a finite number of frames, following the inverted sync symbol.

[0123] FIG. 7 shows a flow chart 700 depicting an embodiment of a process in which a Fast Seamless Rate

Adaptive (FSRA) transmission bit rate change is initiated by a transmitter according to the principles of the invention. In FIG. 7, the steps described in action boxes 710 through 780 correspond to the preceding discussion.

[0124] The FSRA protocol can be completed very quickly. It requires only the exchange of two messages (FSRA grant and FSRA Request) and an inverted sync symbol. FSRA is faster than NSRA because the BAT is stored and need not be exchanged. As in the NSRA protocol, the FSRA protocol is also very robust in noisy environments since it uses inverted sync symbols for the "SRA Go".

[0125] Use of SRA Protocols for Power Management (Entering and Exiting Low Power Modes)

[0126] Full power mode is used during normal operations of the transceiver. Low power transmission modes are often used in transceivers in order to conserve power in cases when data does not need to be transmitted over the line. Many modems have low power modes or "sleep" modes that enable a transceiver to operate at a significantly lower power level when the transmission requirements are reduced. Many modems also have protocols that enable them to enter and exit these low power modes very quickly so that the user is not negatively effected by the modem's transition into the low power mode state. The SRA protocols provided of the invention are used to enter and exit from low power modes in a very fast and seamless manner.

[0127] There are two basic types of low power mode (LPM):

Low Data Rate LPM

[0128] This is low power mode with a very low data rate (e.g. 32 kbps). Only a few of the subchannels are active. The data connection is maintained. The pilot tone may also be transmitted in order to maintain loop timing.

Zero Data Rate LPM

[0129] This is a low power mode with an effectively 0 kbps data rate, i.e., no subchannels are modulating data. A data connection is not maintained. The pilot tone may also be transmitted in this case in order to maintain loop timing.

[0130] In both the Low Data Rate LPM and the Zero Data Rate LPM, the sync symbol, which is sent in normal full power mode every 69 symbols, may be on or off. If the sync symbol is still transmitted during the low power mode, the receiver can use the sync symbol to monitor for channel changes and other fluctuations on the line. However transmission of the sync symbol every 69 symbols can cause non-stationary crosstalk and could be detrimental to other signals on the same telephone wire or in the same wire bundle. If the sync symbol is not transmitted during low power mode, there is no non-stationary crosstalk on the telephone wire or the wire bundle. However, in this case the receiver is not able to monitor the channel with the sync symbol.

[0131] Entering Low Power Mode Using FSRA

[0132] 1. Receiver-initiated Transition into Low Power Mode

[0133] The receiver initiates the transition to low power mode using the receiver-initiated FSRA protocol. A receiver

US 2003/0112884 A1

Jun. 19, 2003

9

initiating the transition to low power mode uses a stored BAT corresponding to the low power mode. The stored BAT table for the low power mode may enable either a Low Data Rate LPM or a Zero Data Rate LPM. The low power mode BAT can be predefined by the system or can be exchanged and stored using the NSRA process. In either case the receiver uses the receiver-initiated FSRA protocol to designate the low power mode BAT and synchronously switch to using that BAT for transmission.

[0134] 2. Transmitter-initiated Transition into Low Power Mode

[0135] There are two ways the transmitter can use the transmitter-initiated FSRA protocol to enter into the low power mode. In one embodiment, the transmitter can use the entire transmitter-initiated FSRA process and request the transition. As in the case of receiver-initiated transition into low power mode, transmitter initiating the transition to low power mode uses a stored BAT for the low power mode. The stored BAT table for the low power mode can enable either a Low Data Rate LPM or a Zero Data Rate LPM. The low power mode BAT can be predefined by the system or can be exchanged and stored using the NSRA process. In either case the transmitter uses the transmitter-initiated FSRA protocol to designate the low power mode BAT and synchronously switches to the low power mode using that BAT for transmission.

[0136] In a second embodiment, the transmitter can transition directly to step 7 of the transmitter initiated FSRA protocol described above, and send the inverted sync symbol to indicate transition into the low power mode. The receiver detects the inverted sync and transitions to the low power mode. In this case, since an FSRA request has not been sent by the transmitter, the receiver recognizes that an inverted sync symbol received without a FSRA request transmitted indicates that the transmitter is switching to low power mode. The low power mode BAT is (predefined by the system) or is identified and stored previously so that both the transmitter and the receiver use the BAT. In an alternative second embodiment, in step 7 the transmitter sends a different signal that is predefined by the transmitter and the receiver to be the signal used for transition into low power mode without an "FSRA request." For example, the transmitter may send a sync symbol with 45 degree phase rotation, rather than the inverted (180 degree) sync symbol. A sync symbol with a 45 degree phase rotation indicates that the transmitter is transitioning into low power mode using the stored BAT associated with the low power mode on the first frame, or a finite number of frames, following the sync symbol with a 45 degree rotation.

[0137] The transmitter-initiated entry into low power mode as defined in the second embodiment has the advantage that it does not require the reverse channel to make the transition. The reverse channel is defined as the communications channel in the opposite direction, i.e., here, the communications channel used to send the FSRA messages from the receiver to the transmitter. This is advantageous because the reverse channel may already be in low power mode with no data connection. If there is no data ready to be sent the transmitter can simply transition to low power mode. This is an important power savings technique since the transmitter consumes a large portion of the power, as it is required to send the signal down the line. Transmitter-

initiated transition into low power modes is also useful in "soft modem" (PC host based) implementations. In a soft modem implementation, the host processor is performing the modem transceiver functions and many other PC applications at the same time. If the host processor must perform another task that does not allow it to run the ADSL transmitter, the processor can quickly transition the transmitter to the low power mode by sending the inverted sync symbol, or the sync symbol with 45 degree rotation. After this the host processor resources can be consumed by the other task. The ADSL transmitter sends no signal (0 kbps) onto the line.

[0138] The transmitter-initiated and receiver-initiated protocols described above enable the communication system to enter a low power mode in each direction (upstream and downstream) separately or in both directions together. The cases described above each focus on one direction. The protocols can be combined to accomplish transition in both directions at the same time. As an example, assume that the customer premise transceiver (CPT) is designed to enter into a low power mode in response to a PC that is also entering a similar state. The CPT first uses receiver-initiated low power mode transition to put the downstream (CO to CPT direction) into low power mode. Afterwards the CPT uses the transmitter-initiated low power mode transition to put the upstream (CPT to CO direction) into low power mode.

[0139] Exiting Low Power Mode

[0140] 1. Receiver-initiated Exit from Power Mode

[0141] According to the SRA protocols, there are two embodiments the receiver can use to exit the low power mode-. In the first embodiment, receiver-initiated exit from low power mode can be accomplished using the receiver initiated NSRA or FSRA protocol if the low power mode still has at least a slow data connection in the reverse direction (Low data rate LPM). This is necessary because the receiver must be capable of sending the SRA request back to the transmitter along with the BAT to be used. If the transmitter has not turned off the sync symbol in low power mode the NSRA or FSRA protocols would be used as described above. If the transmitter sync symbol is turned off while in low power mode, the "SRA Go" is sent by the transmitter by turning the sync symbol back on. The receiver detects the presence of the sync symbol (with or without inversion) as a flag to synchronize the change in data rate.

[0142] In a second embodiment, there is no data connection in the reverse direction (Zero Data Rate LPM). The receiver initiates an exit by first completing a "transmitter-initiated exit from low power mode" (described below) in the reverse direction. This enables the data connection in the reverse direction. The receiver uses the receiver initiated NSRA or FSRA protocol to exit from low power mode in its own direction. As described above, if the transmitter sync symbol is turned off while in low power mode, the "SRA Go" is sent by the transmitter by turning the sync symbol back on. The receiver detects the presence of the sync symbol (with or without inversion) as a flag to synchronize the change in data rate.

[0143] 2. Transmitter-initiated Exit from Low Power Mode

[0144] According to the SRA protocols, there are two embodiments the transmitter can use to exit from low power mode. In the first embodiment, the transmitter uses the entire

US 2003/0112884 A1

Jun. 19, 2003

10

transmitter initiated FSRA or NSRA process and requests the transition. This requires that there is a data connection in both directions (Low data rate LPM) so the protocol messages can be exchanged. As in the receiver-initiated exit from low power mode, if the transmitter has not turned off the sync symbol in low power mode the NSRA or FSRA protocols would be used as described above. If the transmitter had turned the sync symbol off while in low power mode, then the "SRA Go" is sent by the transmitter by turning the sync symbol back on. The receiver detects the presence of the sync symbol (with or without inversion) as a flag to synchronize the change in data rate.

[0145] In the second embodiment, the transmitter can exit the low power mode by transitioning directly to step 7 of the transmitter initiated FSRA protocol. The transmitter sends the inverted sync symbol to indicate transition out of the low power mode. This requires that a sync symbol be sent during the low power mode. This protocol does not require a low data rate LPM. The receiver detects the inverted sync and exits the low power mode. The receiver is designed to recognize that an inverted sync symbol received without a FSRA request indicates the transmitter is exiting from low power mode. The full power mode BAT is identified and stored previously in the connection so that both the transmitter and the receiver have the BAT. For example, the BAT to be used upon exiting a low power mode can be defined by the system to default to the BAT of the last full power connection. Alternatively, the transmitter can send a different signal that is predefined by the transmitter and the receiver to be the signal used for transition out of low power mode without an "FSRA request". For example, the transmitter can send a sync symbol with 45 degree phase rotation, rather than the inverted (180 degree) sync symbol. When the receiver detects the sync symbol with a 45 degree phase rotation, the receiver recognizes that the transmitter is transitioning out of low power mode using the stored BAT associated with the full power mode on the first frame, or a finite number of frames, following the sync symbol with a 45 degree rotation. If the transmitter had turned the sync symbol off while in low power mode, then the "SRA Go" is sent by the transmitter by turning the sync symbol back on. The receiver detects the presence of the sync symbol (with or without a phase shift) as a flag to synchronize the change in data rate.

[0146] Although throughout this description the BAT is defined to be a table that specifies the number of bits allocated to each subchannel, the BAT can also contain other parameters associated with allocating bits to subchannels in a multicarrier system. An example of an additional parameter is the Fine Gain per subchannel as defined in the ANSI and ITU standards. In this case, when the BAT is exchanged during the NSRA protocol or the BAT is stored during the FSRA protocol, the BAT also contains the Fine Gain value for each subchannel.

[0147] The seamless rate adaptive system and associated protocols also applies to DMT systems that implement dual (or multiple) latency paths. A dual latency system is defined in the ITU and ANSI standards as a DMT system that supports two data streams with different latency specifications in the Framing/FEC block. FIG. 3 shows a standard ADSL DMT system 300 that implements dual latency, as an example of a system having a plurality of latencies. The system 300 includes three layers: the Modulation layer 310,

the Framing/FEC layer 320, and the ATM TC layer 340, which are similar but not identical to the three layers described above in relation to FIG. 1.

[0148] The Modulation layer 310 provides functionality associated with DMT modulation. The DMT modulation is implemented using an Inverse Discrete Fourier Transform (IDFT) 112. The IDFT 112 modulates bits from the dual input Quadrature Amplitude Modulation (QAM) 314 encoder into the multicarrier subchannels. The operation of the Modulation layer 310 is analogous to that of Modulation layer 110 of FIG. 1, with the difference that the Modulation layer 310 has multiple inputs, rather than only one input.

[0149] The Framing/FEC layer 320 shown in FIG. 3 has two paths. This layer contains a first path that includes the same blocks as in the Framing/FEC layer 120 of FIG. 1, namely the Interleaving (INJ) block 122, the Forward Error Correction (FEC) block 124, the scrambler (SCR) block 126, the Cyclic Redundancy Check (CRC) block 128 and the ADSL Framing block 130. The layer further contains a second path that includes a second one of each of the Forward Error Correction (FEC) block 124', the scrambler (SCR) block 126', the Cyclic Redundancy Check (CRC) block 128' and the ADSL Framing block 130'. The Framing/FEC layer 320 provides functionality associated with preparing a stream of bits for modulation,

[0150] The new lower path through the Framing/FEC layer 320 has a different amount of latency than the original upper path corresponding to FIG. 1, because the lower path does not perform interleaving on the data stream. Dual latency is used to send different application bit streams with different latency requirements through the ADSL DMT modem. As an example, an application that can tolerate high latency (e.g., video on demand) may be sent through the upper high latency path with interleaving whereas the an application with low latency requirements (e.g., voice) may be sent through the lower low latency path without interleaving.

[0151] The ATM TC layer 340 includes an ATM TC block 342 having multiple inputs and multiple outputs that transforms bits and bytes in cells into frames for each path.

[0152] The seamless rate adaptation system and method of the present invention applies to a system with dual latency, or even multiple latency, as well. In the case of dual latency, the FEC and interleaving parameters for both paths are decoupled from the DMT symbol size. The BAT contains, in addition to the number of bits allocated to each subchannel, the data rate for each latency path in the form of bits per DMT symbol. When seamless rate adaptations are performed using the FSRA and NSRA protocols the BAT also indicates the data rate for each latency path. For example, if the dual latency system runs with 1.536 Mbps on the interleaved path (high latency upper path) and 256 kbps in the non-interleaved path (low latency lower path) and an SRA is initiated, then the SRA protocol specifies the new BAT containing the number of bits per subchannel and also the new data rate for each latency path. At a 4 kHz DMT symbol rate, a system running at 1.536 Mbps+256 kbps=1.792 Mbps $1792000/4000=448$ total bits per symbol. The BAT specifies that $1536000/4000=384$ bits per symbol are allocated to the interleaved path and $256000/4000=64$ bits per symbol are allocated to the non-interleaved path. In the example, when an SRA is performed, the new data rate for the interleaved path can be 1.048 Mbps ($1048000/4000=262$

US 2003/0112884 A1

Jun. 19, 2003

11

bits per symbol) and the new data rate for the non-interleaved path can be 128 kbps ($128000/4000=32$ bits per DMT symbol), resulting in a total throughput rate of 1.176 kbps (or 294 total bits per DMT symbol). The NSRA and FSRA protocols combined with the framing method specified herein complete this data rate change in both latency paths in a seamless manner. It is also possible to not change the data rate on both latency paths. For example one may want to keep the 256 kbps low latency path at a constant data rate because it is carrying voice data (multiple telephone calls) that can not operate at a lower rate, whereas the 1.536 Mbps path may be carrying internet access data that can tolerate a rate change. In this example, during the SRA the data rate of the low latency path is kept constant at 256 kbps whereas the data rate of the high latency path changes.

[0153] While the invention has been disclosed in connection to ADSL systems it can also be applied to any system that utilizes multicarrier modulation. In general this invention applies to any system in which different numbers of bits are modulated on the carriers.

[0154] While the invention has been particularly shown and described with reference to specific preferred embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

What is claimed is:

1. In a multicarrier transmission system including a transmitter and a receiver, the transmitter and receiver using a first bit allocation table to transmit a plurality of codewords at a first transmission bit rate in a first power mode, the plurality of codewords having a specified codeword size and including a specified number of parity bits for forward error correction, and a specified interleaving parameter for interleaving the plurality of codewords, a method for seamlessly entering a second power mode, the method comprising:

storing a second bit allocation table at the receiver and at the transmitter for transmitting codewords at a second transmission in the second power mode;

synchronizing use of the second bit allocation table between the transmitter and receiver; and

entering the second power mode by using the second bit allocation table to transmit codewords,

wherein the specified interleaving parameter, the specified codeword size, and the specified number of parity bits for forward error correction used to transmit codewords in the first power mode are also used to transmit codewords in the second power mode to achieve a seamless change in power mode.

* * * * *

A193 – A195

NOT USED

Thomas Starr | Massimo Sorbara | John M. Cioffi | Peter J. Silverman

DSL Advances

- ADSL, HDSL2, SHDSL, VDSL, line unbundling, spectrum management, and more
- Voice over DSL and video-on-demand: technical progress and its impact on services
- DSL in the home: automating deployment, enhancing security, maximizing revenue
- Improved coding techniques, multi-user detection, and other new advances

Prentice Hall Communications Engineering and Emerging Technologies Series

Theodore S. Rappaport, Series Editor

DSL ADVANCES

Thomas Starr

Massimo Sorbara

John M. Cioffi

Peter J. Silverman



PRENTICE HALL
Professional Technical Reference
Upper Saddle River, New Jersey 07458
www.phptr.com

Library of Congress Cataloging-in-Publication Data

DSL advances / Thomas Starr ... [et al].
p. cm. — (Communications engineering and emerging technologies series)
Includes bibliographical references and index.
ISBN 0-13-093810-6
1. Digital subscriber lines. I. Starr, Thomas. II. Prentice Hall communications engineering and emerging technologies series. TK5103.78 .D75 2003
621.382'3—dc21

2002153605

Editorial/production supervision: *Jessica Balch (Pine Tree Composition, Inc.)*
Cover design director: *Jerry Votta*
Art director: *Gail Cocker-Bogusz*
Interior design: *Meg Van Arsdale*
Manufacturing buyer: *Maura Zaldivar*
Publisher: *Bernard Goodwin*
Marketing manager: *Dan DePasquale*
Editorial assistant: *Michelle Vincenti*
Full-service production manager: *Anne R. Garcia*



© 2003 by Pearson Education, Inc.
Publishing as Prentice Hall Professional Technical Reference
Upper Saddle River, NJ 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

For information regarding corporate and government bulk discounts, please contact: Corporate and Government Sales (800) 382-3419 or corpsales@pearsontechgroup.com

Other company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN: 0-13-093810-6

Pearson Education Ltd., *London*
Pearson Education Australia Pty, Limited, *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Ltd., *Toronto*
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education—Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.

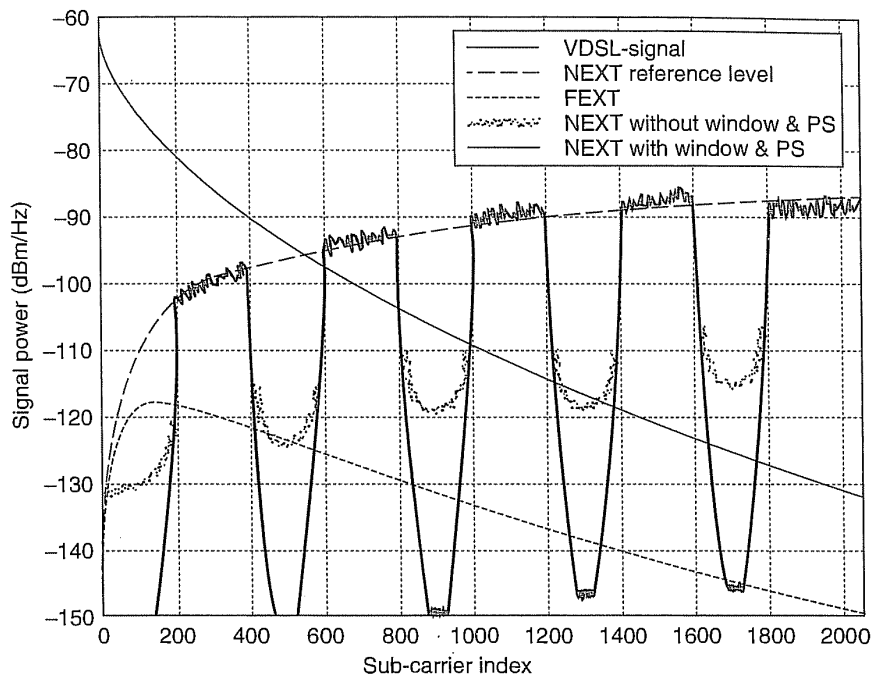


Figure 7.20 PS = transmitter windowing (pulse shaping), and window here means receiver window. This simulation is for a 1000 m loop of .5 mm transmission line (24 gauge).

ceiver, but a dual configuration exists for the upstream receiver. Note any small residual upstream VDSL NEXT left after windowing in the downstream tone n (or in tones less than n in frequency index) must be a function of the upstream signal extracted at frequencies $n + 1, n + 2, \dots$ at the FFT output. This function is a function of the frequency offsets between all the NEXTs and the VDSL signal. This timing-clock offset is usually fixed but can drift with time slowly. An adaptive filter can eliminate the NEXT as per standard noise cancellation methods [6]. A very small number of tones are required for the canceler per up/down edge if transmit windowing and receiver windowing are used. Adaptive noise cancellation can be used to make VDSL self-NEXT negligible with respect to the -140 dBm/Hz noise level. This allows full benefit of any FEXT reduction methods that may also be also in effect (note the NEXT is already below the FEXT even without the NEXT canceler, but reducing it below the noise floor anticipates a VDSL system's potential ability to eliminate or dramatically reduce FEXT). See, for instance, Chapter 11. Note this noise canceller is an early version of the per-tone equalizer concept in [37].

7.3.3 DMT VDSL Framing

The DMT transmission format supports use of Reed-Solomon forward error correction (see [6]) and convolutional/triangular interleaving. The RS code is the same as that used for ADSL with up to 16 bytes of overhead allowed per code word.

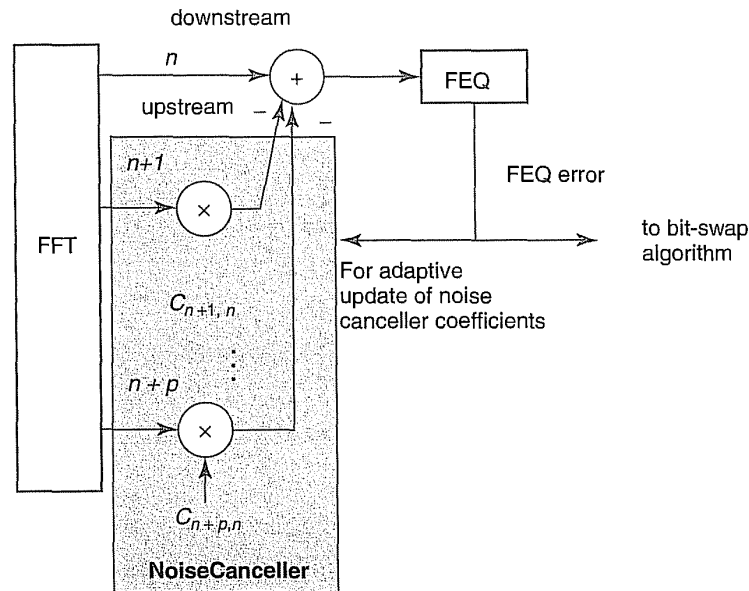


Figure 7.21 Adaptive noise canceler for elimination of VDSL self-NEXT in asynchronous VDSL operation. Shown for one upstream/downstream boundary tone (can be replicated for each up/down transition tone that has NEXT distortion with asynchronous VDSL NEXT). No canceler necessary if NEXT is synchronous.

There is no fixed relationship between symbol boundaries and code word boundaries, unlike ADSL.

Instead, any payload data rate that is an integer multiple of 64 kbps (implying an even integer multiple of payload bytes on average per symbol) with dummy byte insertion where necessary and as described in [17] is allowed. Triangular interleaving that allows interleaving at a block length that is any integer sub-multiple of a code word length (in bytes) is allowed (ADSL forced the block size of the interleaver to be equal to the code word length). Given the high speeds of VDSL, the loss caused by dummy insertion is small, compared with the implementation advantage of decoupling symbol length from code word length. Triangular interleaving was described in [6] and again in [17], so it is not described here.

Latency can take on any value between 1 ms (fast buffer requirement) and 10 ms (slow buffer default) or more. The latency is determined according to code word, data rate, and interleave-depth parameter choices as in [17]. Fast and slow data are combined according to a frame format that no longer includes the synchronization symbol of ADSL and has updates of the fast and slow control bytes with respect to ADSL. Superframes are no longer restricted to just sixty-nine symbols as in ADSL.

Framing can be implemented as in Chapter 9 of [17] and need not be repeated here.

7.3.4 Initialization

An earlier text describes in detail the various aspects of training of a DMT modem [6]. The VDSL training procedure is described in [17] and compatible with the popular g.hs (g. 994) of ITU. The fundamental steps of training are the same as in [6] with the LT now being expected to set a timing advance and measure round-trip delay of signals so that the digital-duplexing becomes automatic. The length of cyclic prefix versus suffix and other detailed framing parameters are set through various initialization exchanges.

One feature of digital duplexing is that it does allow very simple echo cancellation if there is band overlap. With synchronized symbols, there is only one tap per tone to do full echo cancellation where that may be appropriate. However, the NEXT generated by overlapping bands at high frequencies might discourage one from trying unless NEXT cancellation (coordinated transmitters and receivers) can also be used, which would only be one tap per tone per significant crosstalk.

The reader is otherwise referred to [17] for more details.

7.4 MULTIPLE-QAM APPROACHES AND STANDARDS

The VDSL system specified in [16] uses either CAP or QAM as a modulation scheme [6], and analog frequency division duplexing (FDD) to separate the upstream and downstream channels. There are two carriers or equivalently center frequencies, both with 20 percent excess bandwidth raised cosine transmission in each direction, following frequency plans 997 (Europe) or 998 (North America). The symbol rate of each of the signals is any integer multiple of 67.5 kHz, and the carrier/center frequencies can also be programmed as any integer multiple of 33.75 kHz. This allows for the receiver for each signal to estimate signal quality and request an appropriate center frequency and symbol rate, as well as corresponding signal constellation, which can be any integer QAM constellation from four points to 256 points as described in detail in [16]. Radio frequency emission control occurs through programmable notch filters in the transmitter, for which a decision feedback equalizer in the receiver can partially compensate.

With overhead included, data rates are certain integer multiples (not all) of 64 kbps up to 51.84 Mbps downstream and 25.92 Mbps upstream. SCM advocates sometimes cite “blind operation”—that is, no training, as an advantage, but of course one data rate must be fixed for all systems for true blind operation.

7.4.1 Profiling in SCM VDSL

To accommodate short (< 1 kft), medium (1000–3000 feet) and long (> 3000 feet) transmission at both asymmetric and symmetric rates, SCM VDSL can transmit up to four QAM signals, two in each direction. For long loops, only one carrier downstream and one carrier upstream (just above the downstream band) are permitted. For medium range, a second downstream carrier is permitted in addition to the two carriers of long loops, while the short loops can use all four carriers. It was basically this four-max carrier restriction of SCM that forced the number of bands in the 997

ETSI TS 101 270-2 V1.2.1 (2003-07)

Technical Specification

**Transmission and Multiplexing (TM);
Access transmission systems on metallic access cables;
Very High Speed Digital Subscriber Line (VDSL);
Part 2: Transceiver specification**



Reference

RTS/TM-06023

Keywords

access, ATM, B-ISDN, IP, network, transmission,
VDSL, XDSL

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, send your comment to:

editor@etsi.org

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2003.
All rights reserved.

DECTTM, PLUGTESTSTM and UMTSTM are Trade Marks of ETSI registered for the benefit of its Members.
TIPHONTM and the TIPHON logo are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

6.4.1 Scrambler

A scrambler shall be used to reduce the likelihood that a long sequence of zeros will be transmitted over the channel. The scrambler shall be self-synchronising so that descrambling can occur without requiring a particular alignment with the scrambled sequence. The scrambler is represented by the equation below where $m(n)$ is a message bit at time n and the output of the scrambler is $x(n)$:

$$x(n) = m(n) + x(n-18) + x(n-23),$$

all arithmetic is modulo 2. As long as the scrambler is initialised with values other than zero, an "all zeros" sequence for $m(n)$ will result in a pseudo random sequence of length $2^{23} - 1$. At the input to the scrambler, the LSB of each octet enters the scrambler first. At the output of the scrambler, the LSB of each octet leaves the scrambler first.

6.4.2 Forward error correction

A standard octet-oriented Reed-Solomon code shall be used to provide protection against random and burst errors. Comprised of R redundant check octets $c_0, c_1, \dots, c_{R-2}, c_{R-1}$ appended to K message octets $m_0, m_1, \dots, m_{K-2}, m_{K-1}$, a Reed-Solomon code word contains $N=K+R$ octets. The check octets are computed from the message octets using the equation:

$$C(D) = M(D)D^R \bmod G(D),$$

where

$$M(D) = m_0 D^{K-1} \oplus m_1 D^{K-2} \oplus \dots \oplus m_{K-2} D \oplus m_{K-1}$$

is the message polynomial,

$$C(D) = c_0 D^{R-1} \oplus c_1 D^{R-2} \oplus \dots \oplus c_{R-2} D \oplus c_{R-1}$$

is the check polynomial, and $G(D) = \prod (D \oplus \alpha^i)$ is the generator polynomial of the Reed-Solomon code, where the index of the product runs from $i = 0$ to $R-1$. That is, $C(D)$ is the remainder obtained from dividing $M(D)D^R$ by $G(D)$. The arithmetic is performed in the Galois Field GF(256), where α is a primitive element that satisfies the primitive binary polynomial $x^8 \oplus x^4 \oplus x^3 \oplus x^2 \oplus 1$. A data octet $(d_7, d_6, \dots, d_1, d_0)$ is identified with the Galois Field element $d_7 \alpha^7 \oplus d_6 \alpha^6 \oplus \dots \oplus d_1 \alpha \oplus d_0$.

Both K and R are programmable parameters. Redundancy values of $R = 0, 2, 4, 6, 8 \dots 16$ shall be supported. The following code word parameters specified as (N, K) shall be supported: (144, 128) and (240, 224). Other values for N and K are optional. However, N shall be less than or equal to 255.

6.4.3 Interleaving

6.4.3.1 General

Interleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The interleaver and de-interleaver shall be adjustable via the management system to meet latency requirements. The latency of the slow path is a function of the data rate and burst error correction capability. For data rates greater than or equal to 13 Mbps, the latency between the α and β interfaces shall not exceed 10 ms when the interleaver delay is set to the maximum. At lower data rates there is a trade-off between higher latency and decreased burst error correction ability. At any data rate, the minimum latency occurs when the interleaver is turned off.

When the interleaver is on, the codewords shall be interleaved before transmission to increase the immunity of RS codewords to bursts of errors. The convolutional interleaver is defined by two parameters: the interleaver block length, I , and the interleaving depth, D . The block length I divides the RS codeword length N . The convolutional interleaver uses a memory in which a block of I octets is written while an (interleaved) block of I octets is read.

The convolutional interleaver introduces a read-to-write delay, Δ_I , that increments linearly with the octet index within a block of I octets:

$$\Delta_I = (D-1) \times j, \text{ where } j = 0, 1, 2, \dots, I-1.$$

6.4.3.2 Triangular implementation

To decrease the implementation complexity, the delay increment $(D-I)$ shall be chosen as a multiple of the interleaver block length (I) . Therefore, $M \times I = (D-I)$, where the parameter M is an integer. The characteristics of convolutional interleaving are shown in Annex C. Table 19 summarizes interleaving depth, interleaving (and de-interleaving) memory size, and end-to-end delay. The correction capability is calculated using t = number of octets that can be corrected by RS codewords which equals half the number of redundancy octets ($R/2$) and q = length of RS codeword divided by the length of an interleaved block (N/I).

Table 19: Characteristics of triangular, convolutional interleaver

Parameter	Value
Interleaver block length	I octets (I must divide N)
Interleaving depth D	$M \times I + 1$ octets
(De)interleaver memory size	$M \times I \times (I-1)/2$ octets
End-to-end delay	$M \times I \times (I-1)$ octets
Correction capability	$\lfloor t/q \rfloor \times M \times (I+1)$ octets

6.4.4 Framing

6.4.4.1 Frame description

A *frame* is a set of octets carried by one DMT symbol. The frame frequency depends on the length of the cyclic extension. A frame is composed of two sources: the "fast" buffer and the "interleaved" buffer. The index i refers to parameters related to the fast or interleaved buffers ($i \in \{F, I\}$). The inclusion of the fast buffer is optional. When the fast buffer is not included, the interleaved buffer can carry non-interleaved data by setting the interleaver depth to zero.

Both the fast and interleaved buffers contain an integer number of RS-encoded octets. Neither the fast nor the interleaved buffer is required to carry an integer number of RS codewords. To reduce the end-to-end delay, it is recommended that the fast buffer (or the interleaved buffer when the interleaver depth is zero) carries at least one RS codeword. The framing parameters are exchanged between the VTU-O and VTU-R during initialisation.



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.992.1

(06/99)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA,
DIGITAL SYSTEMS AND NETWORKS

Digital transmission systems – Digital sections and digital
line system – Access networks

**Asymmetric digital subscriber line (ADSL)
transceivers**

ITU-T Recommendation G.992.1

(Previously CCITT Recommendation)

ITU-T G-SERIES RECOMMENDATIONS
TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
INTERNATIONAL ANALOGUE CARRIER SYSTEM	
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TESTING EQUIPMENTS	
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TRANSMISSION SYSTEMS	
TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
General	G.900–G.909
Parameters for optical fibre cable systems	G.910–G.919
Digital sections at hierarchical bit rates based on a bit rate of 2048 kbit/s	G.920–G.929
Digital line transmission systems on cable at non-hierarchical bit rates	G.930–G.939
Digital line systems provided by FDM transmission bearers	G.940–G.949
Digital line systems	G.950–G.959
Digital section and digital transmission systems for customer access to ISDN	G.960–G.969
Optical fibre submarine cable systems	G.970–G.979
Optical line systems for local and access networks	G.980–G.989
Access networks	G.990–G.999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION G.992.1**ASYMMETRIC DIGITAL SUBSCRIBER LINE (ADSL) TRANSCEIVERS****Summary**

This Recommendation describes Asymmetric Digital Subscriber Line (ADSL) Transceivers on a metallic twisted pair that allows high-speed data transmission between the network operator end (ATU-C) and the customer end (ATU-R). This Recommendation provides a variety of bearer channels in conjunction with one of three other services dependent on the environment:

- 1) ADSL transmission simultaneously on the same pair with voice (band) service;
- 2) ADSL transmission simultaneously on the same pair with G.961 (Appendix I or II) ISDN services; or
- 3) ADSL transmission on the same pair with voiceband transmission and with TCM-ISDN (G.961 Appendix III) in an adjacent pair.

Systems allow approximately 6 Mbit/s downstream and approximately 640 kbit/s upstream data rates depending on the deployment and noise environment.

This Recommendation specifies the physical layer characteristics of the Asymmetric Digital Subscriber Line (ADSL) interface to metallic loops.

This Recommendation has been written to help ensure the proper interfacing and interworking of ADSL transmission units at the customer end (ATU-R) and at the network operator end (ATU-C) and also to define the transport capability of the units. Proper operation shall be ensured when these two units are manufactured and provided independently. A single twisted pair of telephone wires is used to connect the ATU-C to the ATU-R. The ADSL transmission units must deal with a variety of wire pair characteristics and typical impairments (e.g. crosstalk and noise).

An ADSL transmission unit can simultaneously convey all of the following: downstream simplex bearers, duplex bearers, a baseband analogue duplex channel, and ADSL line overhead for framing, error control, operations and maintenance. Systems support a minimum of 6.144 Mbit/s downstream and 640 kbit/s upstream.

This Recommendation includes mandatory requirements, recommendations and options; these are designated by the words "shall", "should" and "may" respectively. The word "will" is used only to designate events that take place under some defined set of circumstances.

Two categories of performance are specified. Category I performance is required for compliance with this Recommendation; performance enhancement options are not required for category I equipment. Category II is a higher level of performance (i.e. longer lines and greater impairments). Category II performance and characteristics are not required for compliance with this Recommendation.

This Recommendation defines several optional capabilities and features:

- echo cancellation;
- trellis coded modulation;
- dual latency;
- transport of a network timing reference;
- transport of STM and/or ATM;
- reduced overhead framing modes.

It is the intention of this Recommendation to provide, by negotiation during initialization, for U-interface compatibility and interoperability between transceivers complying to this Recommendation and between transceivers that include different combinations of options.

Source

ITU-T Recommendation G.992.1 was prepared by ITU-T Study Group 15 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on 22 June 1999.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2000

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	Page
1 Scope.....	1
1.1 System reference model	2
1.2 Objectives	2
2 References.....	3
3 Definitions	3
4 Abbreviations.....	6
5 Reference Models	9
5.1 ATU-C transmitter reference models.....	9
5.1.1 ATU-C transmitter reference model for STM transport.....	10
5.1.2 ATU-C transmitter reference model for ATM transport	11
5.2 ATU-R transmitter reference models.....	12
5.2.1 ATU-R transmitter reference model for STM transport.....	12
5.2.2 ATU-R transmitter reference model for ATM transport	13
5.3 ATU-C/R transmitter timing model (Annex C only).....	13
6 Transport capacity.....	14
6.1 Transport of STM data.....	15
6.2 Transport of ATM data	16
6.3 ADSL system overheads and total bit rates	17
7 ATU-C functional characteristics	18
7.1 STM Transmission Protocol Specific functionalities	19
7.1.1 ATU-C input and output V interfaces for STM transport	19
7.1.2 Downstream simplex channels – bit rates	19
7.1.3 Downstream/upstream duplex channels – bit rates	19
7.1.4 Payload transfer delay.....	20
7.1.5 Framing Structure for STM transport.....	20
7.2 ATM Transport Protocol Specific functionalities.....	20
7.2.1 ATU-C input and output V interface for ATM transport	20
7.2.2 Payload transfer delay.....	21
7.2.3 ATM Cell specific functionalities	21
7.2.4 Framing Structure for ATM transport	23
7.3 Network timing reference (NTR).....	24
7.3.1 Need for NTR	24
7.3.2 Transport of the NTR	24
7.4 Framing.....	25
7.4.1 Data symbols	25

	Page
7.4.2 Synchronization	33
7.4.3 Reduced overhead framing	35
7.5 Scramblers	37
7.6 Forward error correction	38
7.6.1 Reed-Solomon coding	38
7.6.2 Reed-Solomon Forward Error Correction Superframe Synchronization.....	38
7.6.3 Interleaving	38
7.6.4 Support of higher downstream bit rates with $S = 1/2$ (optional).....	39
7.7 Tone ordering.....	40
7.8 Constellation encoder (trellis code version)	42
7.8.1 Bit extraction	42
7.8.2 Bit conversion.....	43
7.8.3 Coset partition and trellis diagram.....	43
7.8.4 Constellation encoder	47
7.9 Constellation encoder – No trellis coding.....	50
7.9.1 Bit extraction	50
7.9.2 Constellation encoder	50
7.10 Gain scaling	50
7.11 Modulation.....	51
7.11.1 Subcarriers	51
7.11.2 Modulation by the inverse discrete Fourier transform (IDFT)	51
7.11.3 Synchronization symbol	52
7.11.4 Inverse synchronization symbol (Annex C only)	52
7.12 Cyclic prefix.....	53
7.13 Transmitter dynamic range	53
7.13.1 Maximum clipping rate	53
7.13.2 Noise/distortion floor.....	53
7.14 ATU-C Downstream Transmitter spectral masks.....	54
7.15 Dual Bitmapping and Rate Conversion (Annex C only)	54
7.16 FEXT Bitmapping (Annex C only).....	54
8 ATU-R Functional Characteristics	54
8.1 STM Transmission Protocol Specific functionalities	54
8.1.1 ATU-R input and output T interfaces for STM transport.....	54
8.1.2 Downstream simplex channels – Transceiver bit rates.....	55
8.1.3 Duplex channels – Transceiver bit rates.....	55
8.1.4 Framing Structure for STM transport.....	55
8.2 ATM Transport Protocol Specific functionalities.....	56
8.2.1 ATU-R input and output T interfaces for ATM transport	56

	Page
8.2.2 ATM Cell specific functionalities	56
8.2.3 Framing Structure for ATM transport	56
8.3 Network timing reference	57
8.4 Framing	57
8.4.1 Data symbols	57
8.4.2 Synchronization	61
8.4.3 Reduced overhead framing	63
8.5 Scramblers	63
8.6 Forward error correction	63
8.7 Tone ordering	63
8.8 Constellation encoder – Trellis version	64
8.9 Constellation encoder – Uncoded version	64
8.10 Gain scaling	64
8.11 Modulation	64
8.11.1 Subcarriers	64
8.11.2 Modulation by the inverse discrete Fourier transform	65
8.11.3 Synchronization symbol	65
8.11.4 Inverse synchronization symbol (Annex C only)	65
8.12 Cyclic prefix	65
8.13 Transmitter dynamic range	65
8.13.1 Maximum clipping rate	65
8.13.2 Noise/distortion floor	65
8.14 ATU-R Upstream Transmitter spectral response	66
8.15 Dual Bitmapping and Rate Conversion (Annex C only)	66
8.16 FEXT Bitmapping (Annex C only)	66
9 EOC Operations and Maintenance	66
9.1 Clear EOC	66
9.2 Embedded operations channel (EOC) requirements	67
9.2.1 EOC organization and protocol	67
9.2.2 EOC message structure	67
9.2.3 EOC message sets	68
9.2.4 Data registers in the ATU-R	72
9.2.5 EOC protocol states	73
9.3 In-service performance monitoring and surveillance	80
9.3.1 ADSL line related primitives	81
9.3.2 STM data path related primitives	83
9.3.3 ATM data path related primitives	83
9.3.4 Other ADSL indicators, parameters and signals	85

	Page
9.4 Failures and Performance Parameters.....	85
9.5 Test Parameters.....	85
9.5.1 Near-end test parameters	85
9.5.2 Far-end test parameters.....	86
10 Initialization	86
10.1 Overview.....	86
10.1.1 Basic functions of initialization.....	86
10.1.2 Transparency to methods of separating upstream and downstream signals ..	87
10.1.3 Implementation of service options for ADSL.....	87
10.1.4 Resetting during initialization and data transmission.....	87
10.1.5 Initialization with Hyperframe (Annex C only).....	88
10.2 Handshake – ATU-C.....	88
10.2.1 CL messages	88
10.2.2 MS messages	88
10.3 Handshake – ATU-R.....	89
10.3.1 CLR messages	89
10.3.2 MS messages	90
10.4 Transceiver training – ATU-C.....	91
10.4.1 C-QUIET2	91
10.4.2 C-PILOT1	91
10.4.3 C-PILOT1A	92
10.4.4 C-QUIET3A	92
10.4.5 C-REVERB1	92
10.4.6 C-PILOT2	93
10.4.7 C-ECT.....	93
10.4.8 C-REVERB2	93
10.4.9 C-QUIET5	93
10.4.10 C-PILOT3	93
10.4.11 C-REVERB3	93
10.5 Transceiver training – ATU-R.....	93
10.5.1 R-QUIET2	93
10.5.2 R-REVERB1	94
10.5.3 R-QUIET3	94
10.5.4 R-ECT.....	94
10.5.5 R-REVERB2	95
10.6 Channel analysis (ATU-C).....	95
10.6.1 C-SEGUE1	95
10.6.2 C-RATES1.....	95
10.6.3 C-CRC1	96

	Page
10.6.4 C-MSG1.....	97
10.6.5 C-CRC2	98
10.6.6 C-MEDLEY.....	98
10.7 Channel analysis (ATU-R).....	99
10.7.1 R-SEGUE1	99
10.7.2 R-REVERB3	99
10.7.3 R-SEGUE2	99
10.7.4 R-RATES1.....	99
10.7.5 R-CRC1	100
10.7.6 R-MSG1.....	100
10.7.7 R-CRC2	101
10.7.8 R-MEDLEY.....	101
10.7.9 R-REVERB4	102
10.8 Exchange – ATU-C.....	103
10.8.1 C-REVERB4	104
10.8.2 C-SEGUE2	104
10.8.3 C-RATES-RA.....	104
10.8.4 C-CRC-RA1	105
10.8.5 C-MSG-RA.....	106
10.8.6 C-CRC-RA2	106
10.8.7 C-REVERB-RA.....	106
10.8.8 C-SEGUE-RA	106
10.8.9 C-MSG2.....	106
10.8.10 C-CRC3	108
10.8.11 C-RATES2.....	108
10.8.12 C-CRC4	109
10.8.13 C-B&G.....	109
10.8.14 C-CRC5	109
10.8.15 C-REVERB5	109
10.8.16 C-SEGUE3	110
10.9 Exchange – ATU-R.....	110
10.9.1 R-SEGUE3	110
10.9.2 R-MSG-RA.....	110
10.9.3 R-CRC-RA1	111
10.9.4 R-RATES-RA.....	112
10.9.5 R-CRC-RA2	112
10.9.6 R-REVERB-RA.....	112
10.9.7 R-SEGUE-RA	112
10.9.8 R-MSG2.....	113
10.9.9 R-CRC3	114

	Page
10.9.10 R-RATES2.....	114
10.9.11 R-CRC4	114
10.9.12 R-REVERB5	115
10.9.13 R-SEGUE4	115
10.9.14 R-B&G.....	115
10.9.15 R-CRC5	116
10.9.16 R-REVERB6	116
10.9.17 R-SEGUE5	116
10.9.18 Details of Initialization timing.....	116
11 AOC on-line adaptation and reconfiguration.....	119
11.1 The ADSL overhead control (AOC) channel	119
11.1.1 AOC message header.....	119
11.1.2 AOC protocol	119
11.2 On-line adaptation – Bit swapping	120
11.2.1 Bit swap channel.....	120
11.2.2 Superframe counting.....	120
11.2.3 Bit swap request.....	120
11.2.4 Extended bit swap request	121
11.2.5 Bit swap acknowledge	122
11.2.6 Bit swap – Receiver	122
11.2.7 Bit swap – Transmitter	123
Annex A – Specific requirements for an ADSL system operating in the frequency band above POTS	123
A.1 ATU-C Functional Characteristics (pertains to clause 7)	123
A.1.1 Pilot Frequency (supplements 7.11.1.2)	123
A.1.2 ATU-C downstream transmit spectral mask (replaces 7.14).....	123
A.1.3 ATU-C transmitter PSD mask for reduced NEXT	127
A.2 ATU-R (see clause 8).....	128
A.2.1 Modulation by the inverse discrete Fourier transform (see 8.11.2).....	128
A.2.2 Synchronization symbol (supplements 8.11.3).....	129
A.2.3 Cyclic prefix (replaces 8.12).....	129
A.2.4 ATU-R transmitter spectral mask (replaces 8.14)	129
A.2.5 Nyquist frequency (supplements 8.11.1.2)	132
A.3 Initialization (see clause 10)	133
A.3.1 Power cut-back (supplements 10.4.5.1).....	133
A.3.2 Estimated average upstream loop attenuation (see 10.8.9.1).....	133
A.3.3 Estimated average downstream loop attenuation (supplements 10.9.8.1).....	133
A.3.4 C-PILOT1 (supplements 10.4.2)	133
A.3.5 R-REVERB1 (see 10.5.2).....	133

	Page
A.3.6 C-ECT (see 10.4.7).....	133
A.3.7 R-ECT (see 10.5.4).....	133
A.3.8 C-MSG2 (supplements 10.8.9).....	134
A.3.9 R-MSG2 (supplements 10.9.8).....	134
A.4 Electrical characteristics (new).....	134
A.4.1 DC characteristics.....	134
A.4.2 Voiceband characteristics	134
A.4.3 ADSL band characteristics	135
Annex B – Specific requirements for an ADSL system operating in the frequency band above ISDN as defined in ITU-T Recommendation G.961 Appendices I and II.....	136
B.1 ATU Functional Characteristics (see clause 7).....	136
B.1.1 Data Subcarriers (see 7.11.1.1).....	136
B.1.2 Pilot Frequency (see 7.11.1.2)	136
B.1.3 Downstream transmit spectral mask (replaces 7.14)	136
B.2 ATU-R functional characteristics (see clause 8).....	140
B.2.1 ATU-R transmit reference models.....	140
B.2.2 ATU-R Upstream transmit spectral mask (replaces 8.14).....	141
B.2.3 Data subcarriers (replaces 8.11.1.1).....	143
B.2.4 Pilot frequency.....	144
B.2.5 Nyquist frequency (supplements 8.11.1.2)	144
B.2.6 Modulation by the Inverse Discrete Fourier Transform (replaces 8.11.2).....	144
B.2.7 Synchronization symbol (supplements 8.11.3).....	144
B.2.8 Cyclic prefix (replaces 8.12).....	144
B.3 Initialization (see clause 10)	145
B.3.1 Handshake – ATU-C (supplements 10.2).....	145
B.3.2 Handshake – ATU-R (supplements 10.3).....	145
B.3.3 Power Cut-back (supplements 10.4.5.1).....	146
B.3.4 Estimated average upstream loop attenuation (see 10.8.9.1).....	146
B.3.5 Estimated average downstream loop attenuation (10.9.8.1).....	146
B.3.6 C-PILOT1 (for 10.4.3).....	146
B.3.7 R-REVERB1 (supplements 10.5.2).....	146
B.3.8 R-MEDLEY (replaces 10.7.8).....	147
B.3.9 C-MSG2 (supplements 10.8.9).....	147
B.3.10 R-MSG2 (supplements 10.9.8).....	147
B.3.11 C-ECT and R-ECT (supplements 10.4.7 and 10.5.4).....	147
B.3.12 C-B&G (replaces 10.8.13).....	147
Annex C – Specific requirements for an ADSL system operating in the same cable as ISDN as defined in ITU-T Recommendation G.961 Appendix III.....	150
C.1 Scope.....	150

	Page
C.2 Definitions	150
C.3 Reference Models	151
C.3.1 ATU-C transmitter reference model (replaces figures in 5.1)	151
C.3.2 ATU-R transmitter reference model (replaces figures in 5.2)	153
C.3.3 ATU-C/R transmitter timing model (replacement for 5.3)	155
C.4 ATU-C functional characteristics (pertains to clause 7)	158
C.4.1 STM transmission protocols specific functionality (pertains to 7.1)	158
C.4.2 ATM transmission protocols specific functionalities (pertains to 7.2)	159
C.4.3 Framing (pertains to 7.4)	159
C.4.4 Dual Bitmapping and Rate Conversion (replaces 7.15)	164
C.4.5 FEXT Bitmapping (replaces 7.16)	166
C.4.6 Tone Ordering (replacement for 7.7)	167
C.4.7 Modulation (pertains to 7.11)	167
C.4.8 ATU-C Downstream transmit spectral mask (replaces 7.14)	167
C.5 ATU-R Functional Characteristics (pertains to clause 8)	168
C.5.1 Framing (pertains to 8.4)	168
C.5.2 Dual Bitmapping and rate conversion (replaces 8.15)	171
C.5.3 FEXT Bitmapping (replaces 8.16)	173
C.5.4 Tone Ordering (pertains to 8.7)	173
C.5.5 Modulation (pertains to 8.11)	173
C.5.6 ATU-R Upstream Transmit Spectral Mask (supplements 8.14)	174
C.6 EOC Operation and Maintenance (pertains to clause 9)	174
C.6.1 ADSL line related primitives (supplements 9.3.1)	174
C.6.2 Test Parameters (supplements 9.5)	174
C.7 Initialization	174
C.7.1 Initialization with Hyperframe (replaces 10.1.5)	174
C.7.2 Handshake – ATU-C (supplements 10.2)	178
C.7.3 Handshake – ATU-R (supplements 10.3)	178
C.7.4 Transceiver Training – ATU-C (supplements 10.4)	179
C.7.5 Transceiver Training – ATU-R (supplements 10.5)	180
C.7.6 Channel analysis (ATU-C) (supplements 10.6)	181
C.7.7 Channel analysis (ATU-R) (supplements 10.7)	183
C.7.8 R-SEGUE1 (supplements 10.7.1)	184
C.7.9 Exchange – ATU-C (supplements 10.8)	185
C.7.10 Exchange – ATU-R (supplements 10.9)	186
C.8 AOC On-line adaptation and reconfiguration (pertains to clause 11)	191
Annex D – ATU-C and ATU-R state diagrams	192
D.1 Introduction	192
D.2 Definitions	192

	Page
D.3 State Diagrams	193
Annex E – POTS and ISDN-BA splitters	198
E.1 Type 1 – European	198
E.1.1 Impedance matching	198
E.1.2 Return loss	198
E.1.3 Insertion loss	199
E.1.4 Insertion loss distortion	199
E.1.5 Isolation	199
E.1.6 Signal power	199
E.1.7 Ringing	199
E.1.8 Unbalance about Earth (ETS-300.001, § 4.2.1 and § 4.2.2).....	199
E.1.9 Frequencies and levels for pulse metering (ETS-300.001, § 1.7.8).....	200
E.2 Type 2 – North America	200
E.2.1 Introduction.....	200
E.2.2 DC characteristics.....	203
E.2.3 Voiceband characteristics	203
E.2.4 ADSL band testing	209
E.2.5 Home premises physical considerations	211
E.3 Type 3 – ADSL above ISDN (ITU-T G.961 Appendix I or II).....	211
E.4 Type 4 – Type for Japan.....	212
E.4.1 Introduction.....	212
E.4.2 DC Characteristics	212
E.4.3 AC Characteristics	214
Annex F – ATU-x Classification and Performance for Region A (North America)	219
F.1 Classification definitions	219
F.2 Performance requirements	219
F.2.1 Crosstalk interference	220
F.2.2 Impulse noise	221
F.2.3 POTS	222
Annex G – ATU-x Classification and Performance for Region B (Europe)	223
G.1 Performance requirements	223
Appendix I – ATM Layer to Physical Layer Logical Interface	224
Appendix II – Dynamic (on-line) Rate Adaptation.....	226
II.1 Introduction.....	226
II.1.1 General concepts.....	227

	Page
II.2 DRA protocol and messages	227
II.2.1 DRA concept	227
II.2.2 DRA-AOC messages	227
II.3 Monitoring	228
II.3.1 DRA_Monitor_Request	228
II.3.2 DRA_Monitor_Reply	228
II.4 Configuration	229
II.4.1 DRA_Configuration_Request	229
II.4.2 DRA_Configuration_Reply	230
II.5 Exchange	231
II.5.1 DRA_Exchange_Request	231
II.5.2 DRA_Exchange_Reply	232
II.6 Swap	233
II.6.1 Example	235
II.6.2 DRA_Swap_Request	235
II.6.3 DRA_Swap_Reply	236
II.7 DRA state diagram	237
II.7.1 State machine conventions	237
II.7.2 ATU-R state machine	238
II.7.3 ATU-C state machine	239
Appendix III – Compatibility with other customer premises equipment	240
Appendix IV – Bibliography	240

7.6 Forward error correction

The ATU-C shall support downstream transmission with at least any combination of the FEC coding capabilities shown in Table 7-7.

Table 7-7/G.992.1 – Minimum FEC coding capabilities for ATU-C

Parameter	Fast buffer	Interleaved buffer
Parity bytes per R-S codeword	$R_F = 0, 2, 4, 6, 8, 10, 12, 14, 16$ (Note 1)	$R_I = 0, 2, 4, 6, 8, 10, 12, 14, 16$ (Notes 1 and 2)
DMT symbols per R-S codeword	$S = 1$	$S = 1, 2, 4, 8, 16$
Interleave depth	Not applicable	$D = 1, 2, 4, 8, 16, 32, 64$
NOTE 1 – R_F can be > 0 only if $K_F > 0$, and R_I can be > 0 only if $K_I > 0$.		
NOTE 2 – R_I shall be an integer multiple of S .		

The ATU-C shall also support upstream transmission with at least any combination of the FEC coding capabilities shown in Table 8-3.

7.6.1 Reed-Solomon coding

R (i.e. R_F or R_I) redundant check bytes $c_0, c_1, \dots, c_{R-2}, c_{R-1}$ shall be appended to K (i.e. K_F or $S \times K_I$) message bytes $m_0, m_1, \dots, m_{K-2}, m_{K-1}$ to form a Reed-Solomon codeword of size $N = K + R$ bytes. The check bytes are computed from the message byte using the equation:

$$C(D) = M(D) D^R \text{ modulo } G(D) \quad (7-16)$$

where:

$$M(D) = m_0 D^{K-1} + m_1 D^{K-2} + \dots + m_{K-2} D + m_{K-1} \quad (7-17)$$

is the message polynomial,

$$C(D) = c_0 D^{R-1} + c_1 D^{R-2} + \dots + c_{R-2} D + c_{R-1} \quad (7-18)$$

is the check polynomial, and

$$G(D) = \prod_{i=0}^{R-1} (D + \alpha^i) \quad (7-19)$$

is the generator polynomial of the Reed-Solomon code, where the index of the product runs from $i = 0$ to $R-1$. That is, $C(D)$ is the remainder obtained from dividing $M(D) D^R$ by $G(D)$. The arithmetic is performed in the Galois Field GF(256), where α is a primitive element that satisfies the primitive binary polynomial $x^8 + x^4 + x^3 + x^2 + 1$. A data byte ($d_7, d_6, \dots, d_1, d_0$) is identified with the Galois Field element $d_7\alpha^7 + d_6\alpha^6 + \dots + d_1\alpha + d_0$.

The number of check bytes R , and the codeword size N vary, as explained in 7.4.

7.6.2 Reed-Solomon Forward Error Correction Superframe Synchronization

When entering the SHOWTIME state after completion of Initialization and Fast Retrain, the ATU shall align the first byte of the first Reed-Solomon codeword with the first data byte of DF 0.

7.6.3 Interleaving

The Reed-Solomon codewords in the interleave buffer shall be convolutionally interleaved. The interleaving depth varies, as explained in 7.4, but shall always be a power of 2. Convolutional

interleaving is defined by the rule:

Each of the N bytes B_0, B_1, \dots, B_{N-1} in a Reed-Solomon codeword is delayed by an amount that varies linearly with the byte index. More precisely, byte D_i (with index i) is delayed by $(D-1) \times i$ bytes, where D is the interleave depth.

An example for $N = 5, D = 2$ is shown in Table 7-8, where B_{ji} denotes the i -th byte of the j -th codeword.

Table 7-8/G.992.1 – Convolutional interleaving example for $N = 5, D = 2$

Interleaver input	B_{j0}	B_{j1}	B_{j2}	B_{j3}	B_{j4}	B_{j+1_0}	B_{j+1_1}	B_{j+1_2}	B_{j+1_3}	B_{j+1_4}
Interleaver output	B_{j0}	B_{j-1_3}	B_{j1}	B_{j-1_4}	B_{j2}	B_{j+1_0}	B_{j3}	B_{j+1_1}	B_{j4}	B_{j+1_2}

With the above-defined rule, and the chosen interleaving depths (powers of 2), the output bytes from the interleaver always occupy distinct time slots when N is odd. When N is even, a dummy byte shall be added at the beginning of the codeword at the input to the interleaver. The resultant odd-length codeword is then convolutionally interleaved, and the dummy byte shall then be removed from the output of the interleaver.

7.6.4 Support of higher downstream bit rates with $S = 1/2$ (optional)

With a rate of 4000 data frames per second and a maximum of 255 bytes (maximum RS codeword size) per data frame, the ADSL downstream line rate is limited to approximately 8 Mbit/s per latency path. The line rate limit can be increased to about 16 Mbit/s for the interleaved path by mapping two RS codewords into one FEC data frame (i.e. by using $S = 1/2$ in the interleaved path). $S = 1/2$ shall be used in the downstream direction only over bearer channel AS0. Support of $S = 1/2$ is optional.

When the K_I data bytes per interleaved mux data frame cannot be packed into one RS codeword, i.e. K_I is such that $K_I + R > 255$, the K_I data bytes shall be split into two consecutive RS codewords. When K_I is even, the first and second codeword have the same length $N_{I1} = N_{I2} = (K_I/2 + R_I)$, otherwise the first codeword is one byte longer than the second, i.e. first codeword has $N_{I1} = (K_I + 1)/2 + R_I$ bytes, the second codeword has $N_{I2} = (K_I - 1)/2 + R_I$ bytes. For the FEC output data frame, $N_I = N_{I1} + N_{I2}$, with $N_I < 511$ bytes.

The convolutional interleaver requires all codewords to have the same odd length. To achieve the odd codeword length, insertion of a dummy (not transmitted) byte may be required. For $S = 1/2$, the dummy byte addition to the first and/or second codeword at the input of the interleaver shall be as in Table 7-9.

Table 7-9/G.992.1 –Dummy byte insertion at interleaver input for $S = 1/2$

N_{Id1}	N_{Id2}	Dummy Byte Insertion Action
Odd	Odd	No action
Even	Even	Add one dummy byte at the beginning of both codewords
Odd	Even	Add one dummy byte at the beginning of the second codeword
Even	Odd	Add one dummy byte at the beginning of the first codeword and two dummy bytes at the beginning of the second codeword [the de-interleaver shall insert one dummy byte into the de-interleaver matrix on the first byte and the $(D + 1)$ th byte of the corresponding codeword to make the addressing work properly]



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.992.3

(07/2002)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA,
DIGITAL SYSTEMS AND NETWORKS

Digital sections and digital line system – Access networks

**Asymmetric digital subscriber line
transceivers 2 (ADSL2)**

ITU-T Recommendation G.992.3

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TESTING EQUIPMENTS	G.500–G.599
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
General	G.900–G.909
Parameters for optical fibre cable systems	G.910–G.919
Digital sections at hierarchical bit rates based on a bit rate of 2048 kbit/s	G.920–G.929
Digital line transmission systems on cable at non-hierarchical bit rates	G.930–G.939
Digital line systems provided by FDM transmission bearers	G.940–G.949
Digital line systems	G.950–G.959
Digital section and digital transmission systems for customer access to ISDN	G.960–G.969
Optical fibre submarine cable systems	G.970–G.979
Optical line systems for local and access networks	G.980–G.989
Access networks	G.990–G.999
QUALITY OF SERVICE AND PERFORMANCE	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DIGITAL TERMINAL EQUIPMENTS	G.7000–G.7999
DIGITAL NETWORKS	G.8000–G.8999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation G.992.3

Asymmetric digital subscriber line transceivers 2 (ADSL2)

Summary

This Recommendation describes Asymmetric Digital Subscriber Line (ADSL) Transceivers on a metallic twisted pair that allows high-speed data transmission between the network operator end (ATU-C) and the customer end (ATU-R). It defines a variety of frame bearers in conjunction with one of two other services or without underlying service, dependent on the environment:

- 1) ADSL transmission simultaneously on the same pair with voice band service;
- 2) ADSL transmission simultaneously on the same pair with ISDN (Appendix I or II/G.961 [1]) services;
- 3) ADSL transmission without underlying service, optimized for deployment with ADSL over voiceband service in the same binder cable;
- 4) ADSL transmission without underlying service, optimized for deployment with ADSL over ISDN service in the same binder cable.

ADSL transmission on the same pair with voiceband services and operating in an environment with TCM-ISDN (Appendix III/G.961 [1]) services in an adjacent pair, is for further study.

This Recommendation specifies the physical layer characteristics of the Asymmetric Digital Subscriber Line (ADSL) interface to metallic loops.

This Recommendation has been written to help ensure the proper interfacing and interworking of ADSL transmission units at the customer end (ATU-R) and at the network operator end (ATU-C), and also to define the transport capability of the units. Proper operation shall be ensured when these two units are manufactured and provided independently. A single twisted pair of telephone wires is used to connect the ATU-C to the ATU-R. The ADSL transmission units must deal with a variety of wire pair characteristics and typical impairments (e.g., crosstalk and noise).

An ADSL transmission unit can simultaneously convey all of the following: a number of downstream frame bearers, a number of upstream frame bearers, a baseband POTS/ISDN duplex channel, and ADSL line overhead for framing, error control, operations, and maintenance. Systems support a net data rate ranging up to a minimum of 8 Mbit/s downstream and 800 kbit/s upstream. Support of net data rates above 8 Mbit/s downstream and support of net data rates above 800 kbit/s upstream are optional.

This Recommendation includes mandatory requirements, recommendations and options; these are designated by the words "shall", "should" and "may" respectively. The word "will" is used only to designate events that take place under some defined set of circumstances.

This Recommendation defines several optional capabilities and features:

- transport of STM and/or ATM and/or Packets;
- transport of a network timing reference;
- multiple latency paths;
- multiple frame bearers;
- short initialization procedure;
- dynamic rate repartitioning;
- seamless rate adaptation.

It is the intention of this Recommendation to provide, by negotiation during initialization, for U-interface compatibility and interoperability between transceivers complying with this Recommendation and between transceivers that include different combinations of options.

History

This Recommendation describes the second generation of ADSL, based on the first generation ITU-T Rec. G.992.1. It is intended that this Recommendation be implemented in multi-mode devices that support both ITU-T Recs G.992.3 and G.992.1.

This Recommendation has been written to provide additional features, relative to ITU-T Rec. G.992.1. ITU-T Rec. G.992.1 was approved in June 1999. Since then, several potential improvements have been identified in areas such as data rate versus loop reach performance, loop diagnostics, deployment from remote cabinets, spectrum control, power control, robustness against loop impairments and RFI, and operations and maintenance. This Recommendation provides a new ADSL U-interface specification, including the identified improvements, which the ITU-T believes will be most helpful to the ADSL industry.

Relative to ITU-T Rec. G.992.1, the following application-related features have been added:

- Improved application support for an all digital mode of operation and voice over ADSL operation;
- Packet TPS-TC function, in addition to the existing STM and ATM TPS-TC functions;
- Mandatory support of 8 Mbit/s downstream and 800 kbit/s upstream for TPS-TC function #0 and frame bearer #0;
- Support for IMA in the ATM TPS-TC;
- Improved configuration capability for each TPS-TC with configuration of latency, BER and minimum, maximum and reserved data rate.

Relative to ITU-T Rec. G.992.1, the following PMS-TC-related features have been added:

- A more flexible framing, including support for up to 4 frame bearers, 4 latency paths;
- Parameters allowing enhanced configuration of the overhead channel;
- Frame structure with receiver selected coding parameters;
- Frame structure with optimized use of RS coding gain;
- Frame structure with configurable latency and bit error ratio;
- OAM protocol to retrieve more detailed performance monitoring information;
- Enhanced on-line reconfiguration capabilities including dynamic rate repartitioning.

Relative to ITU-T Rec. G.992.1, the following PMD-related features have been added:

- New line diagnostics procedures available for both successful and unsuccessful initialization scenarios, loop characterization and troubleshooting;
- Enhanced on-line reconfiguration capabilities including bitswaps and seamless rate adaptation;
- Optional short initialization sequence for recovery from errors or fast resumption of operation;
- Optional seamless rate adaptation with line rate changes during showtime;
- Improved robustness against bridged taps with receiver determined pilot tone;
- Improved transceiver training with exchange of detailed transmit signal characteristics;
- Improved SNR measurement during channel analysis;
- Subcarrier blackout to allow RFI measurement during initialization and SHOWTIME;
- Improved performance with mandatory support of trellis coding;
- Improved performance with mandatory one-bit constellations;
- Improved performance with data modulated on the pilot tone;
- Improved RFI robustness with receiver determined tone ordering;
- Improved transmit power cutback possibilities at both CO and remote side;
- Improved Initialization with receiver and transmitter controlled duration of initialization states;
- Improved Initialization with receiver-determined carriers for modulation of messages;
- Improved channel identification capability with spectral shaping during Channel Discovery and Transceiver Training;
- Mandatory transmit power reduction to minimize excess margin under management layer control;
- Power saving feature for the central office ATU with new L2 low power state;
- Power saving feature with new L3 idle state;
- Spectrum control with individual tone masking under operator control through CO-MIB;
- Improved conformance testing including increase in data rates for many existing tests.

Through negotiation during initialization, the capability of equipment to support the G.992.3 and/or the G.992.1 Recommendations is identified. For reasons of interoperability, equipment may choose to support both Recommendations, such that it is able to adapt to the operating mode supported by the far-end equipment.

Source

ITU-T Recommendation G.992.3 was approved by ITU-T Study Group 15 (2001-2004) under the ITU-T Recommendation A.8 procedure on 29 July 2003.

It integrates the modifications introduced by ITU-T Rec. G.992.3 (2002) Amendment 1 approved on 22 May 2003.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2003

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

	Page
1 Scope.....	1
2 References.....	2
3 Definitions	3
4 Abbreviations.....	6
5 Reference models.....	8
5.1 ATU functional model.....	8
5.2 User plane protocol reference model.....	10
5.3 Management plane reference model.....	11
5.4 Application models.....	11
6 Transport Protocol Specific Transmission Convergence (TPS-TC) function	16
6.1 Transport capabilities	16
6.2 Interface signals and primitives.....	17
6.3 Control parameters	18
6.4 Data plane procedures	19
6.5 Management plane procedures	19
6.6 Initialization procedure.....	19
6.7 On-line reconfiguration	20
6.8 Power management mode.....	20
7 Physical Media Specific Transmission Convergence (PMS-TC) function	21
7.1 Transport capabilities	21
7.2 Additional functions	23
7.3 Block interface signals and primitives	23
7.4 Block diagram and internal reference point signals	26
7.5 Control parameters	28
7.6 Frame structure.....	29
7.7 Data plane procedures	35
7.8 Control plane procedures.....	38
7.9 Management plane procedures	42
7.10 Initialization procedures	43
7.11 On-line reconfiguration	49
7.12 Power management mode.....	50
8 Physical media dependent function	52
8.1 Transport capabilities	52
8.2 Additional functions	53
8.3 Block interface signals and primitives	54
8.4 Block diagram and internal reference point signals	56
8.5 Control parameters	57
8.6 Constellation encoder for data symbols	67
8.7 Constellation encoder for synchronization and L2 exit symbols	82
8.8 Modulation	84
8.9 Transmitter dynamic range.....	87

	Page
8.10 Transmitter spectral masks	88
8.11 Control plane procedures.....	89
8.12 Management plane procedures	89
8.13 Initialization procedures	98
8.14 Short initialization procedures.....	140
8.15 Loop diagnostics mode procedures	144
8.16 On-line reconfiguration of the PMD function	159
8.17 Power management in the PMD function	161
9 Management Protocol Specific Transmission Convergence (MPS-TC) functions	162
9.1 Transport functions.....	163
9.2 Additional functions	163
9.3 Block interface signals and primitives	163
9.4 Management plane procedures	165
9.5 Power management	185
10 Dynamic behaviour.....	189
10.1 Initialization.....	189
10.2 On-line Reconfiguration (OLR)	189
10.3 Power management	192
Annex A – Specific requirements for an ADSL system operating in the frequency band above POTS	195
A.1 ATU-C functional characteristics (pertains to clause 8)	195
A.2 ATU-R functional characteristics (pertains to clause 8)	198
A.3 Initialization.....	200
A.4 Electrical characteristics	200
Annex B – Specific requirements for an ADSL system operating in the frequency band above ISDN as defined in ITU-T Rec. G.961 Appendices I and II.....	205
B.1 ATU-C functional characteristics (pertains to clause 8)	205
B.2 ATU-R functional characteristics (pertains to clause 8)	209
B.3 Initialization.....	212
B.4 Electrical characteristics	213
Annex C – Specific requirements for an ADSL system operating in the same cable as ISDN as defined in ITU-T Rec. G.961 Appendix III	213
Annex D – ATU-C and ATU-R state diagrams.....	213
D.1 Introduction	213
D.2 Definitions	213
D.3 State diagrams	214
Annex E – POTS and ISDN Basic Access Splitters	221
E.1 Type 1 – POTS splitter – Europe	221
E.2 Type 2 – POTS splitter – North America	221
E.3 Type 3 – ISDN (ITU-T Rec. G.961 Appendix I or II) Splitter – Europe	233
E.4 Type 4 – POTS splitter – Japan	234
Annex F – ATU-x performance requirements for region A (North America).....	246
F.1 Performance requirements for operation of ADSL over POTS (Annex A) ...	246

	Page
F.2 Performance requirements for operation of All Digital Mode ADSL (Annex I)	247
Annex G – ATU-x performance requirements for region B (Europe).....	247
G.1 Performance requirements for operation of ADSL over POTS (Annex A) ...	247
G.2 Performance requirements for operation of ADSL over ISDN (Annex B)....	247
G.3 Performance requirements for operation of All Digital Mode ADSL (Annex I)	248
G.4 Performance requirements for operation of All Digital Mode ADSL (Annex J)	248
Annex H – Specific requirements for a synchronized symmetrical DSL (SSDSL) system operating in the same cable binder as ISDN as defined in ITU-T Rec. G.961 Appendix III.....	248
Annex I – All digital mode ADSL with improved spectral compatibility with ADSL over POTS.....	249
I.1 ATU-C functional characteristics (pertains to clause 8)	249
I.2 ATU-R functional characteristics (pertains to clause 8)	251
I.3 Initialization.....	253
I.4 Electrical characteristics.....	253
Annex J – All Digital Mode ADSL with improved spectral compatibility with ADSL over ISDN	257
J.1 ATU-C functional characteristics (pertains to clause 8)	257
J.2 ATU-R functional characteristics (pertains to clause 8)	258
J.3 Initialization.....	261
J.4 Electrical characteristics.....	261
Annex K – TPS-TC functional descriptions	261
K.1 STM Transmission Convergence (STM-TC) function.....	261
K.2 ATM Transmission Convergence (ATM-TC) function	271
K.3 Packet transmission convergence function (PTM-TC)	284
Appendix I – ATM layer to physical layer logical interface	292
Appendix II – Compatibility with other customer premises equipment.....	294
Appendix III – The impact of primary protection devices on line balance	294
III.1 Scope	294
III.2 Background.....	294
III.3 Recommended maximum capacitance of over-voltage protectors.....	296
III.4 Capacitance matching requirements of over-voltage protectors	296
III.5 References	298
Appendix IV – Bibliography.....	299

The signals shown in Figures 9-3 and 9-4 are used to carry primitives between functions of this Recommendation. Primitives are only intended for purposes of clearly specifying functions to assure interoperability.

The primitives that are used between a G.997.1 function and an MPS-TC function are described in Figure 9-1. These primitives support the exchange of clear eoc and command messages.

The primitives that are used between the MPS-TC and PMS-TC functions are defined in 6.2. The primitives that are used between the MPS-TC and the PMD functions are defined in clause 8.

The primitives used to signal maintenance indication primitives to the local maintenance entity are described in respective clauses for TPS-TC, PMS-TC, and PMD functions, (clauses 6, 7 and 8).

Table 9-1/G.992.3 – Signalling primitives between G.997.1 functions and the MPS-TC function

Signal	Primitive	Description
Management. Cleareoc	.request	The transmit G.997.1 function passes clear eoc messages to the MPS-TC function to be transported with this primitive.
	.confirm	This primitive is used by the transmit MPS-TC function to confirm receipt of a Management.Cleareoc.request primitive. By the interworking of the request and confirm, the data flow is matched to the PMS-TC configuration.
	.indicate	The receive MPS-TC function passes clear eoc messages to the receive G.997.1 function that has been transported with this primitive.
Management. Command	.request	The transmit G.997.1 function at the ATU-C passes a command to the ATU-C transmit MPS-TC function to be transported with this primitive.
	.confirm	This primitive is used by the ATU-C receive MPS-TC function to convey the response of the ATU-R to a command. By the interworking of the request and confirm, data may be read from locations.
	.indicate	The receive ATU-R MPS-TC function passes a command to the local ATU-R that has been transported with this primitive.
	.response	This primitive is used by the local ATU-R to convey the response to a command for transport.

9.4 Management plane procedures

9.4.1 Commands

Commands provide for a generalized command, parameters followed by a response. This provides the necessary flexibility to transport clear eoc messages and G.997.1 MIB elements, to set and query ATU registers, and to invoke management procedures at the far end ATU with and without return values.

All commands are categorized into three priority levels, used to determine the order of transport of messages available to the PMS-TC function. The commands are displayed in Tables 9-2, 9-3 and 9-4 in decreasing level of PMS-TC transport priority.

All ATUs should be able to transmit overhead commands and shall respond to all overhead commands as required during operation in the management plane procedures.

All commands received from Tables 9-2, 9-3 and 9-4 shall have a response, noting that the PMS-TC function will discard improperly framed or formatted messages. The responder shall respond within the timeout period displayed in Table 7-17 (dependent on the overhead command priority) less than 50 ms to prevent protocol glare interaction between the ATUs. Shorter responses are allowed and may be required in some application specific situations outside the scope of this Recommendation.

Table 9-2/G.992.3 – Highest priority overhead messages

Message and designator	Direction	Command content	Response content
On-line Reconfiguration (OLR) Command 0000 0001 _b	From a receiver to the transmitter	New configuration including all necessary PMS-TC and PMD control values.	Followed by either a line signal corresponding to the PMD.Synchflag primitive (not a OLR command) or an OLR command for defer or reject.

Table 9-3/G.992.3 – Normal priority overhead messages

Message and designator	Direction	Command content	Response content
EOC Command 0100 0001 _b	From ATU-C to ATU-R	Self test, update test parameters, start and stop TX corrupt CRC, start and stop receipt of corrupt CRC.	Followed by an eoc command for acknowledge.
	From ATU-R to ATU-C	Update test parameters.	Followed by an eoc command for acknowledge.
Time Command 0100 0010 _b	From ATU-C to ATU-R	Set or read time.	Followed by a set time command for acknowledge or the time response.
Inventory Command 0100 0011 _b	From either ATU to the other	Identification request, Self test request, auxiliary inventory information request, PMD capabilities request, PMS-TC capabilities request, TPS-TC capabilities request.	Followed by an inventory command response that includes ATU equipment ID, auxiliary inventory information, set test results, and capabilities information.
Control Parameter Read Command 0000 0100 _b	From either ATU to the other	PMD settings read, PMS-TC settings read, or TPS-TC settings read.	Followed by a control parameter read command response that includes all control variables.
Management Counter Read Command 0000 0101 _b	From either ATU to the other	Null.	Followed by a management counter read response that includes all counter values.

Table 9-3/G.992.3 – Normal priority overhead messages

Message and designator	Direction	Command content	Response content
Power Management Command 0000 0111 _b	From one ATU to the other	Proposed new power state.	Followed by either a line signal corresponding to the PMD.Synchflag primitive (not a power management command) or a power management command for either reject or grant.
Clear eoc Command 0000 1000 _b	From one ATU to the other	Clear eoc message as defined in ITU-T Rec. G.997.1 or other.	Followed by a clear eoc command for acknowledge.
Non-Standard Facility Command 0011 1111 _b	From one ATU to the other	Non-standard identification field followed by message content.	Followed by a non-standard facility command for either acknowledge or negative acknowledge to indicate whether the non-standard identification field is recognized or not.

Table 9-4/G.992.3 – Low priority overhead messages

Message and designator	Direction	Comment content	Response content
PMD Test Parameter Read Command 1000 0001 _b	From either ATU to the other	Parameter number for single read, parameter number and subcarrier id for multiple read, null for next multiple read.	Followed by a PMD test parameter read command response including the requested test parameters or a negative acknowledge.
Non-Standard Facility Low Priority Command 1011 1111 _b	From one ATU to the other	Non-standard identification field followed by message content.	Followed by a non-standard facility command for either acknowledge or negative acknowledge to indicate if the non-standard identification field is recognized.

In the subclauses of 9.4.1 that follow, the format, protocol, and function of each command is specified. For each command, a table is provided that specifies the format of the command and any associated data. To avoid repetition, the command table does not contain the full HDLC frame structure. Commands shall be mapped into the HDLC structure specified in 7.8.2.3, such that Message length P is the number of octets as shown in the first column of the command table. Octet values shall be mapped such that the least significant bit is mapped into the LSB of the HDLC structure. Values spanning more than one octet shall be mapped with higher order octets preceding lower order octets. A vector of value shall be mapped in order of the index, from the lowest index value to highest. Arrays with two indices shall be mapped by decomposing them into a series of vectors using the first index, from the lowest index to the highest. The following example is intended to clarify the mapping from the command table to the HDLC frame structure specified in 7.8.2.3.

The example selected is that of a receiver sending an OLR command repartition the data rate without modification of the underlying PMD function. For this example, the configuration before and after the OLR command is shown in Table 9-5. The HDLC frame content for this message is shown in Table 9-6 and is based on the command format information in Table 9-7.

Table 9-5/G.992.3 – OLR example configuration

Parameter	Current configuration	Proposed configuration
Number of enabled frame bearers	$N_{BC} = 2$	$N_{BC} = 2$
Number of enabled latency path functions	$N_{LP} = 2$	$N_{LP} = 2$
Bits from each latency path function per PMD primitive	$L_0 = 408$	$L_0 = 312$
	$L_1 = 8$	$L_1 = 104$
Frame bearer octets per mux data frame in each latency paths	$B_{00} = 48, B_{01} = 0$	$B_{00} = 36, B_{01} = 0$
	$B_{10} = 0, B_{11} = 0$	$B_{10} = 0, B_{11} = 12$

Table 9-6/G.992.3 – OLR example HDLC frame contents

Octet #	MSB	LSB
	$7E_{16}$ – Opening Flag	
1	Address Field	
2	Control Field	
3	0000 0001 _b (OLR command)	
4	0000 0010 _b (Request Type 2)	
5	0000 0001 _b (L_0 high octet)	
6	0011 1000 _b (L_0 low octet)	
7	0000 0000 _b (L_1 high octet)	
8	0110 1000 _b (L_1 low octet)	
9	0010 0100 _b (B_{00})	
10	0000 1100 _b (B_{11})	
11	0000 0000 _b (N_f) (Message length $P = 9$)	
12	FCS high octet	
13	FCS low octet	
	$7E_{16}$ – Closing Flag	

9.4.1.1 On-line reconfiguration command

The on-line reconfiguration commands shall be used to control certain on-line dynamic behaviour defined in this clause. Additional information is provided on this dynamic behaviour in clause 10. On-line reconfiguration commands may be initiated by either ATU as shown in Table 9-7. However, the initiator is only provided with means to effect changes in its receiver and the corresponding transmitter. The responding ATU may use the on-line reconfiguration commands shown in Table 9-8 or may positively acknowledge the initiator's request by transmitting a line signal corresponding to the PMD.Synchflag primitive. The on-line reconfiguration commands shall consist of multiple octets. The first octet shall be the on-line reconfiguration command designator shown in Table 9-2. The remaining octets shall be as shown in Tables 9-7, 9-8 and 9-9. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-7/G.992.3 – On-line reconfiguration commands transmitted by the initiating receiver

Message length (Octets)	Element name (Command)
$3 + 3 \times N_f$	01 ₁₆ Request Type 1 followed by: 1 octet for the number of subcarriers N_f $3 \times N_f$ octets describing subcarrier parameter field for each subcarrier
$3 + 2 \times N_{LP} + N_{BC} + 3 \times N_f$	02 ₁₆ Request Type 2 followed by: $2 \times N_{LP}$ octets containing new L_p values for the N_{LP} enabled latency paths, N_{BC} octets containing new $B_{p,n}$ values for the N_{BC} enabled frame bearers, 1 octet for the number of carriers N_f $3 \times N_f$ octets describing subcarrier parameter field for each subcarrier
$3 + 2 \times N_{LP} + N_{BC} + 3 \times N_f$	03 ₁₆ Request Type 3 followed by: $2 \times N_{LP}$ octets containing new L_p values for the N_{LP} enabled latency paths, N_{BC} octets containing new $B_{p,n}$ values for the N_{BC} enabled frame bearers, 1 octet for the number of carriers N_f $3 \times N_f$ octets describing subcarrier parameter field for each subcarrier All other octet values are reserved by the ITU-T.

Table 9-8/G.992.3 – On-line reconfiguration commands transmitted by the responding transmitter

Message length (Octets)	Element name (Command)
3	81 ₁₆ Defer Type 1 Request followed by: 1 octet for reason code
3	82 ₁₆ Reject Type 2 Request followed by: 1 octet for reason code
3	83 ₁₆ Reject Type 3 Request followed by: 1 octet for reason code
	All other octet values are reserved by the ITU-T

An ATU may request only changes in its receiver operation. Changes may be requested concurrently by both ATUs; each transaction shall follow the procedures described in this clause. An ATU-R shall not initiate an OLR command if it has transmitted an L2 Grant command and is awaiting a response.

A subcarrier parameter field contains 3 octets formatted as [cccc cccc gggg gggg gggg bbbb]. The carrier index i (8-bits), the g_i (12 bits) and the b_i (4 bits). The carrier index shall be the first octet of the subcarrier field. The g_i shall be contained in the second octet and the four most significant bits of the third octet. The least significant bits of g_i shall be contained in the third octet. The b_i shall be the least significant 4 bits of the third octet.

Type 1 and Type 2 shall be sent such that the PMD parameter L is unchanged. If an ATU implements the optional short PMD initialization sequence, then the ATU should also implement Type 3 OLR operations changing b_i , g_i and L_p .

Reason codes associated with the OLR commands are shown in Table 9-9.

Table 9-9/G.992.3 – Reason codes for OLR commands

Reason	Octet value	Applicable to defer type 1	Applicable to reject type 2	Applicable to reject type 3
Busy	01 ₁₆	X	X	X
Invalid parameters	02 ₁₆	X	X	X
Not enabled	03 ₁₆		X	X
Not supported	04 ₁₆		X	X

Upon transmitting an on-line reconfiguration command, the initiator shall await a response to the command, either an on-line reconfiguration command for defer or reject or the line signal corresponding to the PMD.Synchflag primitive. If the response is not received within the timeout of the high priority overhead messages displayed in Table 7-17, the initiator shall abandon the current on-line reconfiguration command. A new command may be initiated immediately, including an identical request.

Upon receipt of an on-line reconfiguration command, the responder shall respond with either an on-line reconfiguration command for defer or reject, or the line signal corresponding to the PMD.Synchflag primitive. In the case of sending the line signal corresponding to the PMD.Synchflag primitive, the ATU shall reconfigure the effected PMD, PMS-TC, and TPS-TC functions as described in the reconfiguration clauses describing those functions. In the case of defer or reject, the receiver shall supply a reason code from the following: 01₁₆ for busy, 02₁₆ for invalid parameters, 03₁₆ for not enabled, and 04₁₆ for not supported. The reason code 01₁₆ and 02₁₆ shall be the only codes used in an on-line reconfiguration command for defer type 1 request.

Upon receipt of a line signal corresponding to the PMD.Synchflag primitive, the initiator shall reconfigure the effected PMD, PMS-TC, and TPS-TC functions as described in the reconfiguration clauses describing those functions. If an on-line reconfiguration command for defer or reject is received, the initiator shall abandon the current on-line reconfiguration command. A new command may be initiated immediately, including an identical request.

9.4.1.2 eoc Commands

The eoc commands shall be used to control certain in-use diagnostic capabilities defined in this clause. Most eoc commands may be initiated by the ATU-C as shown in Table 9-10. The ATU-R may only initiate the eoc commands shown in Table 9-11. The eoc command shall consist of 2 octets. The first octet shall be the eoc command designator shown in Table 9-3. The second octet shall be as shown in Tables 9-10 and 9-11. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-10/G.992.3 – eoc Commands transmitted by the ATU-C

Message length (Octets)	Element name (Command)
2	01 ₁₆ Perform Self Test
2	02 ₁₆ Update Test Parameters
2	03 ₁₆ Start TX Corrupt CRC
2	04 ₁₆ End TX Corrupt CRC
2	05 ₁₆ Start RX Corrupt CRC
2	06 ₁₆ End RX Corrupt CRC
2	80 ₁₆ ACK
	All other octet values are reserved by the ITU-T.

Table 9-11/G.992.3 – eoc Commands transmitted by the ATU-R

Message length (Octets)	Element name (Command)
2	02 ₁₆ Update Test Parameters
3	01 ₁₆ Self Test Acknowledge followed by a single octet that indicates the minimum time in seconds to wait before requested the self test result
2	80 ₁₆ ACK All other octet values are reserved by the ITU-T.

The eoc command may be transmitted anytime during the on-line state, including immediately following the end of the initialization procedures.

In all cases, the receipt of the eoc command is acknowledged to the transmitter by an eoc command acknowledge (ACK) message. The receiver shall not send a negative acknowledge (NACK) eoc command.

9.4.1.2.1 Self test

Upon receipt of the eoc command for perform set test, the receiving ATU shall transmit the eoc command for self test acknowledge, including the minimum amount of time to wait until requesting the results of the self-test. The receiving ATU shall then perform a self test procedure and generate a self test result. The duration and specific procedure of the self test are vendor discretionary but they shall not interfere with the functions of the ATU and the status of connections. Therefore, the self test procedure performed upon receipt of this command may differ from those performed in the SELFTEST state shown in Figures D.1 and D.2. The result of the self test shall be stored within the indicated number of seconds of transmitting the ACK message. The indicated amount of time shall be between 1 and 255 s.

The most significant octet of the self test result shall be 00₁₆ if the self test passed and 01₁₆ if it failed. The meaning of "failure" is vendor discretionary. The length of the self test result is 4 octets, and the syntax of all other octets is vendor discretionary.

The result of self test may be accessed using the inventory command defined in 9.4.1.4.

9.4.1.2.2 Update test parameters

Upon receipt of the eoc command for update test parameters, the receiving ATU shall transmit the eoc command ACK message and update the test parameter set as defined in 9.4.1.10. Test parameters shall be updated and stored within 10 s after the request is received. Upon receipt of the eoc command ACK message, the transmitting ATU shall wait at least 10 s after arrival of the eoc command ACK message before starting the overhead commands defined in 9.4.1.10 to access the test parameter values.

Upon receipt of this command, the test parameter values relating to the most recent initialization procedure shall be no longer accessible through the overhead commands defined in 9.4.1.10 within 10 s. They may be discarded by the receiving ATU immediately upon receipt of the eoc command for update test parameters.

9.4.1.2.3 Start/End transmit corrupt CRC

Upon receipt of the eoc command for start transmit corrupt CRC, the receiving ATU PMS-TC function shall transmit the eoc command ACK message and transmit a corrupted CRC value in all latency paths until cancelled by the eoc command for end transmit corrupt CRC. A corrupt CRC is any one that does not correspond to the CRC procedure in 7.7.1.2. Only the CRC value is affected by this eoc command. This command may be used conjunction with the eoc command for receive

corrupt CRC (either previously or subsequently) so that both the transmit and receive CRC values are corrupted. The PMS-TC function of the transmitting ATU shall not be affected by this eoc command.

Upon receipt of the eoc command for end transmit corrupt CRC, the receiving ATU PMS-TC function shall transmit the eoc command ACK message and transmit CRC bits determined by the procedure in 7.7.1.2. This command may be transmitted even if the eoc command for start transmit corrupt CRC has not been transmitted. The PMS-TC function of the transmitting ATU shall not be affected by this eoc command.

9.4.1.2.4 Start/End receive corrupt CRC

Upon receipt of the eoc command for start receive corrupt CRC, the receiving ATU shall send the eoc command ACK message. Upon receipt of that eoc command ACK message, the transmitting ATU PMS-TC function shall begin transmitting corrupt CRC bits in all latency paths until cancelled by the eoc command for end receive corrupt CRC. A corrupt CRC is any one that does not correspond to the CRC procedure in 7.7.1.2. This command may be used conjunction with the eoc command for transmit corrupt CRC (either previously or subsequently) so that both the transmit and receive CRC values are corrupted. The PMS-TC function of the receiving ATU shall not be affected by this eoc command.

Upon receipt of the eoc command for end receive corrupt CRC, the receiving ATU shall transmit the eoc command ACK message. Upon receipt of the eoc command ACK message, the transmitting ATU PMS-TC function shall transmit CRC bits determined by the procedure in 7.7.1.2. This command may be transmitted even if the eoc command for start receive corrupt CRC has not been transmitted. The PMS-TC function of the receiving ATU shall not be affected by this eoc command.

9.4.1.3 Time commands

The ATU-C and ATU-R shall each contain timers that are utilized to maintain performance monitoring counters as described in ITU-T Rec. G.997.1 [4]. It is common practice to correlate the counters on each of the DSL line. To facilitate this, it is necessary to synchronize the timers on each end of the line. The set time and read time commands are provided for this purpose. The counters defined in ITU-T Rec. G.997.1 [4] should be updated each time the time counter contains a time value that is an integer multiple of 15 minutes (e.g., 1:00:00, 3:15:00, 15:30:00, 23:45:00).

The requirements for timer accuracy and drift are under study.

The time commands shall be used to synchronize clocks in the ATU as defined in this clause. The time command may be initiated by the ATU-C as shown in Table 9-12. The ATU-R may only reply using the commands shown in Table 9-13. The time commands shall consist of multiple octets as shown in Tables 9-12 and 9-13. The first octet shall be the time command designator shown in Table 9-3. The following octet shall be as shown in Tables 9-12 and 9-13. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-12/G.992.3 – Time command transmitted by the ATU-C

Message length (Octets)	Element name (Command)
10	01 ₁₆ Set followed by 8 octets formatted as HH:MM:SS per ISO 8601 [5]
2	02 ₁₆ Read
	All other octet values are reserved by the ITU-T.

Table 9-13/G.992.3 – Time commands transmitted by the ATU-R

Message length (Octets)	Element name (Command)
2	80 ₁₆ ACK
10	82 ₁₆ Read followed by 8 octets formatted as HH:MM:SS per ISO 8601 [5] All other octet values are reserved by the ITU-T.

Upon receipt of the set time command, the receiving ATU shall transmit the ACK response message. The receiving ATU shall then set its internal clock to the value contained in the message.

Upon receipt of the read time command, the receiving ATU shall transmit the response message that includes the current value of the time counter.

9.4.1.4 Inventory command

The inventory commands shall be used to determine the identification and capabilities of the far ATU as defined in this clause. The inventory commands may be initiated by either ATU as shown in Table 9-14. The responses shall be using the command shown in Table 9-15. The inventory command shall consist of a two octets. The first octet shall be inventory command designator shown in Table 9-3. The second octet shall be one of the values shown in Table 9-14. The inventory response command shall be multiple octets. The first octet shall be inventory command designator shown in Table 9-3. The second shall be the same as the received inventory command second octet, XOR 80₁₆. The remaining octets shall be as shown in Table 9-15. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-14/G.992.3 – Inventory commands transmitted by the initiator

Message length (Octets)	Element name (Command)
2	01 ₁₆ Identification
2	02 ₁₆ Auxiliary Identification
2	03 ₁₆ Self Test Result
2	04 ₁₆ PMD Capabilities
2	05 ₁₆ PMS-TC Capabilities
2	06 ₁₆ TPS-TC Capabilities
	All other octet values are reserved by the ITU-T.

Table 9-15/G.992.3 – Inventory command transmitted by the responder

Message length (Octets)	Element name (Command)
58	81 ₁₆ followed by: 8 octets of vendor id 16 octets of version number 32 octets of serial number
variable	82 ₁₆ followed by: 8 octets of vendor id and multiple octets of auxiliary inventory information
6	83 ₁₆ followed by: 4 octets of self test results
variable	84 ₁₆ followed by: PMD capabilities information
variable	85 ₁₆ followed by: PMS-TC capabilities information
variable	86 ₁₆ followed by: TPS-TC capabilities information
	All other octet values are reserved by the ITU-T.

Upon receipt of one of the inventory commands, the receiving ATU shall transmit the corresponding response message. The function of the receiving or transmitting ATUs is not otherwise affected.

The vendor ID in the identification response shall be formatted according to the vendor id of G.994.1. The vendor ID field is used to specify the system integrator. In this context, the system integrator usually refers to the vendor of the smallest field-replaceable unit. As such, the vendor ID in this response may not be the same as the vendor ID indicated within G.994.1.

The version number, serial number, and auxiliary inventory information shall be assigned with respect to the same system integrator as contained in the vendor ID. The syntax of these fields is vendor discretionary and may be interpreted based on the vendor id presented.

The self test results shall be the results of the most recent self test procedure, initiated either at power-up or by the eoc command for self test. The results shall formatted as defined in 9.4.1.2.1.

For a receiving ATU-C, the PMD, PMS-TC or TPS-TC capabilities information shall consist of the last previously transmitted G.994.1 CL message, reduced to respectively PMD, PMS-TC, or TPS-TC codepoints only. This is followed by the (Npmd/8) PMD, (Npms/8) PMS-TC or (Ntps/8) TPS-TC octets respectively, included in the last previously transmitted C-MSG1 message (see Table 8-37). Codepoints related to the PMD sublayer are defined in Table 8-20. Codepoints related to the PMS-TC sublayer are defined in Table 7-18. Codepoints related to the TPS-TC sublayer are defined in Table 6-2 and Annex K. The octets shall be transmitted in the same order as they are transmitted in the CL and C-MSG1 message.

For a receiving ATU-R, the PMD, PMS-TC or TPS-TC capabilities information shall consist of the last previously transmitted G.994.1 CLR message, reduced to respectively PMD, PMS-TC, or TPS-TC codepoints only, as defined below. This is followed by the (Npmd/8) PMD, (Npms/8) PMS-TC or (Ntps/8) TPS-TC octets respectively, included in the last previously transmitted R-MSG1 message (see Table 8-38). Codepoints related to the PMD sublayer are defined in Table 8-22. Codepoints related to the PMS-TC sublayer are defined in Table 7-18. Codepoints related to the TPS-TC sublayer are defined in Table 6-2 and Annex K. The octets shall be transmitted in the same order as they are transmitted in the CLR and R-MSG1 message.

A CL or CLR message shall be reduced to information related to a particular sublayer only, while maintaining the G.994.1 tree structure for Par(2) block parsing by the transmitting ATU, through the following steps:

- 1) Take the Standard Information Field Par(2) block, under the currently selected Spar(1);
- 2) Set all Npar(2) and Spar(2) codepoints not related to the sublayer to zero;
- 3) Delete all Npar(3) blocks for which the Spar(2) bit has been set to 0;
- 4) Octets at the end of any Par block that contain all ZEROs except for delimiting bits may be omitted from transmission, provided that terminating bits are correctly set for the transmitted octets (see 9.2.3/G.994.1).

9.4.1.5 Control value read commands

The control parameter commands shall be used to determine the current values of all control parameters within the far ATU as defined in this clause. The control parameter commands may be initiated by either ATU as shown in Table 9-16. The responses shall be using the command shown in Table 9-17. The control parameter command shall consist of two octets. The first octet shall be control parameter command designator shown in Table 9-3. The second octet shall be one of the values shown in Table 9-16. The control parameter response command shall be multiple octets. The first octet shall be control parameter command designator shown in Table 9-3. The second shall be the same as the received control parameter command second octet, XOR 80_{16} . The remaining octets shall be as shown in Table 9-17. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-16/G.992.3 – Control parameter commands transmitted by the initiator

Message length (Octets)	Element name (Command)
2	01_{16} PMD Control Parameters
2	02_{16} PMS-TC Control Parameters
2	03_{16} TPS-TC Control Parameters
	All other octet values are reserved by the ITU-T.

Table 9-17/G.992.3 – Control parameter command transmitted by the responder

Message length (Octets)	Element name (Command)
variable	81_{16} followed by: PMD control parameter values
variable	82_{16} followed by: PMS-TC control parameter values
variable	83_{16} followed by: TPS-TC control parameter values
	All other octet values are reserved by the ITU-T.

Upon receipt of one of the control parameter commands, the receiving ATU shall transmit the corresponding response message. The function of the receiving or transmitting ATUs is not otherwise affected.

The control parameter values contained within the PMD, PMS-TC, and TPS-TC responses shall be the transmit function control parameters currently in use by the receiving ATU.

For a receiving ATU-C, the PMD, PMS-TC or TPS-TC control parameter values shall consist of the last previously transmitted G.994.1 MS message, reduced to respectively PMD, PMS-TC, or TPS-TC codepoints only. Within the PMD control parameters only, this is followed by $(4 + N_{SCds}/8)$ octets in R-MSG-PCB format (see Table 8-32, with parameters as defined below). Then follow the $(N_{pmd}/8)$ PMD, $(N_{pms}/8)$ PMS-TC or $(N_{tps}/8)$ TPS-TC octets respectively, included in the last previously transmitted R-PARAMS message (see Table 8-40), and possibly updated during Showtime. Codepoints related to the PMD sublayer are defined in Table 8-21. Codepoints related to the PMS-TC sublayer are defined in Table 7-19. Codepoints related to the TPS-TC sublayer are defined in Table 6-2 and Annex K. The octets shall be transmitted in the same order as they are transmitted in the MS, R-MSG-PCB and R-PARAMS messages.

The ATU-C shall set the octets related to R-MSG-PCB (see Table 8-32) as follows:

- R-MIN_PCB_DS is set to PCBds;
- R-MIN_PCB_US is set to 0;
- HOOK_STATUS is set to 0;
- C-PILOT is set to the pilot subcarrier index currently used by the ATU-C transmit PMD function;
- R-BLACKOUT bits are set to the values currently used by the ATU-C transmit PMD function;
- Other bits are reserved and set to 0.

For a receiving ATU-R, the PMD, PMS-TC or TPS-TC control parameter values shall consist of the last previously transmitted G.994.1 MS message, reduced to respectively PMD, PMS-TC, or TPS-TC codepoints only. Within the PMD control parameters only, this is followed by $(2 + N_{SCus}/8)$ octets in C-MSG-PCB format (see Table 8-27, with parameters as defined below). Then follow the $(N_{pmd}/8)$ PMD, $(N_{pms}/8)$ PMS-TC or $(N_{tps}/8)$ TPS-TC octets respectively, included in the last previously transmitted C-PARAMS message (see Table 8-39), and possibly updated during Showtime. Codepoints related to the PMD sublayer are defined in Table 8-23. Codepoints related to the PMS-TC sublayer are defined in Table 7-19. Codepoints related to the TPS-TC sublayer are defined in Annex K. The octets shall be transmitted in the same order as they are transmitted in the MS and C-PARAMS messages.

The ATU-R shall set the octets related to C-MSG-PCB (see Table 8-27) as follows:

- C-MIN_PCB_DS is set to 0;
- C-MIN_PCB_US is set to PCBus;
- HOOK_STATUS is set to 0;
- C-BLACKOUT bits are set to the values currently used by the ATU-C transmit PMD function;
- Other bits are reserved and set to 0.

An MS message shall be reduced to information on a particular sublayer only, while maintaining the G.994.1 tree structure for parsing by the transmitting ATU, through the same steps as taken for reducing the CL or CLR message.

9.4.1.6 Management counter read commands

The management counter read commands shall be used to access the value of certain management counters maintained by the far ATU in accordance with ITU-T Rec. G.997.1 [4]. The local counter values for completed time intervals shall be retrieved as described in this clause. The management counter read command may be initiated by either ATU as shown in Table 9-18. The responses shall

be using the command shown in Table 9-19. The management counter read command shall consist of a two octets. The first octet shall be management counter read command designator shown in Table 9-3. The second octet shall be one of the values shown in Table 9-18. The management counter read response command shall be multiple octets. The first octet shall be management counter read command designator shown in Table 9-3. The second shall be the same as the received management counter read command second octet, XOR 80_{16} . The remaining octets shall be as shown in Table 9-19. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-18/G.992.3 – Management counter read commands transmitted by the initiator

Message length (Octets)	Element name (Command)
2	01_{16} All other octet values are reserved by the ITU-T.

Table 9-19/G.992.3 – Management counter read command transmitted by the responder

Message length (Octets)	Element name (Command)
$2 + 4 \times (2 \times N_{LP} + 5)$ for PMS-TC and variable for TPS-TC	81_{16} followed by: all the PMS-TC counter values, followed by all the TPS-TC counter values. All other octet values are reserved by the ITU-T.

Upon receipt of one of the management counter read command, the receiving ATU shall transmit the corresponding response message. The function of the receiving or transmitting ATUs is not otherwise affected.

The management counter values shall be derived according to ITU-T Rec. G.997.1 [4] from locally generated defects and anomalies defined within the clauses 6, 7 and 8. The parameters are transferred in the order and format defined in Table 9-20. The TPS-TC anomaly definitions are dependent upon the TPS-TC type and are defined in the Annex K. All PMD and TPS-TC counter values are defined as 32 bit counters and are inserted in the response message most significant to least significant octet order. For latency paths and TPS-TC functions not currently enabled, no octets shall be inserted into the message.

The counters shall be reset at power-on. The counters shall not be reset with a link state transition and shall not be reset when read. The time periods when the ATU is powered but not in the Showtime state shall be counted as unavailable seconds (see 7.2.1.1.9/G.997.1).

Table 9-20/G.992.3 – ATU management counter values

PMD & PMS-TC
Counter of the FEC-0 anomalies
Counter of the FEC-1 anomalies
Counter of the FEC-2 anomalies
Counter of the FEC-3 anomalies
Counter of the CRC-0 anomalies
Counter of the CRC-1 anomalies
Counter of the CRC-2 anomalies
Counter of the CRC-3 anomalies
FEC errored seconds counter
Errored seconds counter
Severely errored seconds counter
LOS errored seconds counter
Unavailable errored seconds counter
TPS-TC
Counters for TPS-TC #0
Counters for TPS-TC #1
Counters for TPS-TC #3
Counters for TPS-TC #4

9.4.1.7 Power management commands

The power management command shall be used to propose power management transitions from one link state to another as described in the Power management subclause 9.5. The power management command may be initiated by either ATU as prescribed in the Power management subclause 9.5 as shown in Table 9-21. The responses shall be using the command shown in Table 9-22. The power management command is variable in length. The first octet shall be power management command designator shown in Table 9-3. The remaining octets shall be as shown in Table 9-21. The power management response commands are variable in length. The first octet shall be power management command designator shown in Table 9-3. The second shall be as shown in Table 9-22. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-21/G.992.3 – Power management commands transmitted by the initiating ATU

Message length (Octets)	Element name (Command)
3	01 ₁₆ Simple Request followed by: 1 octet for the new proposed link state
$4 + 4 \times N_{LP}$	02 ₁₆ L2 Request followed by: 1 octet for minimum PCBds value (dB) 1 octet for maximum PCBds value (dB) $2 \times N_{LP}$ octets containing maximum L_p values for the N_{LP} enabled latency paths, $2 \times N_{LP}$ octets containing minimum L_p values for the N_{LP} enabled latency paths
3	03 ₁₆ L2 Trim followed by the 1 octet for the proposed new value of PCBds (dB) All other octet values are reserved by the ITU-T.

Table 9-22/G.992.3 – Power management command transmitted by the responding ATU

Message length (Octets)	Element name (Command)
2	80 ₁₆ Grant
3	81 ₁₆ Reject followed by: 1 octet for reason code
$6 + 2 \times N_{LP} + 3 \times N_f$	82 ₁₆ L2 Grant followed by: $2 \times N_{LP}$ octets containing new L_p values for the N_{LP} enabled latency paths, 1 octet containing the actual PCBds value 1 octet containing the exit symbol PCBds value, 1 octet containing the exit symbol b_i/g_i table flag, 1 octet for the number of carriers N_f $3 \times N_f$ octets describing subcarrier parameter field for each subcarrier
3	83 ₁₆ L2 Reject followed by: 1 octet for reason code
3	84 ₁₆ L2 Trim Grant followed by 1 octet containing the exit symbol PCBds value
3	85 ₁₆ L2 Trim Reject followed by: 1 octet for reason code
	All other octet values are reserved by the ITU-T.

In the L2 Request, L2 Grant, and L2 Trim Request, and L2 Trim Grant messages, power cutback values shall be expressed as an absolute power cutback in the range of 0 to 40 dB in steps of 1 dB. The cutback is defined in terms of PCBds. The minimum and maximum requested values are defined in absolute terms and not relative to the current PCBds value. Values not inclusively within the range of the PCBds determined during initialization to 40 dB shall not be encoded. It is intended that up to 40 dB of absolute power cutback can be performed for the L2 link state using the PCBds control parameter and that the gain values can be used to additionally adjust the gain per carrier as required.

Reason codes associated with the power management commands are shown in Table 9-23.

Table 9-23/G.992.3 – Reason codes for power management commands

Reason	Octet value	Applicable to reject	Applicable to L2 reject	Applicable to L2 trim reject
Busy	01 ₁₆	X	X	
Invalid	02 ₁₆	X	X	X
State Not Desired	03 ₁₆	X		
Infeasible Parameters	04 ₁₆		X	X

9.4.1.7.1 Simple request by ATU-R

Upon receipt of the power management simple request command, the responding ATU-C will transmit either the Grant or Reject command. The link state shall be formatted as 00₁₆, and 03₁₆ for L0 and L3 link states, respectively. If any other link state is received, the response shall be the Reject response using reason code 02₁₆. The ATU-C shall follow procedures defined in 9.5.3.5 or 9.5.3.1, depending upon the proposed power state L0 or L3, respectively. The ATU-C may also reject a request to move to link state L3 using reason code 01₁₆ because it is temporarily too busy or

using code 03₁₆ because it has local knowledge that the L3 state is not desired at this time. The ATUs may immediately start the protocol to request transition to the same or a different link state. The ATU-C shall not reject a request to move to link state L0.

In case the ATU-R requests exit from L2 into the L0 state, the ATU-C shall not respond with a Grant command. The ATU-C shall respond with the L2 exit sequence, as defined in 8.7.

9.4.1.7.2 Simple request by ATU-C

Upon receipt of the power management Simple Request command, the responding ATU-R will transmit either the Grant or Reject command. The link state shall be formatted as 03₁₆ for L3 link states. If any other link state is received, the response shall be the Reject response using reason code 02₁₆. The ATU-R shall follow procedures defined in 9.5.3.1 to move to link state L3. The ATU-R may instead reject a request to move to link state L3 using reason code 01₁₆ because it is temporarily too busy or 03₁₆ because it has local knowledge that the L3 state is not desired at this time. The ATUs may immediately start the protocol to request transition to the same or a different link state.

9.4.1.7.3 L2 request by ATU-C

When sending the L2 Request command, the ATU-C shall specify parameters describing the minimum and maximum average power cutback, defined in terms of the PMD control parameter PCBds. The ATU-C shall also specify the minimum and maximum L_p value for each configured PMS-TC latency path function. Values larger than the current L_p values shall not be encoded.

Upon receipt of the L2 Request command, the ATU-R shall evaluate the parameters found in the L2 Request message and the current operating conditions of the downstream receiver. If the parameters are invalid (i.e., not within the allowed encoding ranges), the ATU-R shall send a L2 Reject command using reason code 02₁₆. If the parameters are valid but describe an operating condition that cannot be currently satisfied (e.g., because the current line and noise conditions cannot support the configuration), the ATU-R shall send a L2 Reject command using reason code 04₁₆. If the parameters can be met, the ATU-R shall send an L2 Grant command and follow procedures defined in 9.5.3.3. The L2 Grant command shall contain the actual value of PCBds necessary modifications to the bits and gain tables to be used by the ATUs in the downstream direction. Additionally, the grant command shall describe the PCBds and the b_i/g_i flag value that the ATU-C shall use to transmit a L2 exit sequence as described in 8.7. These should be selected by the receiver to best assure reliable detection of the L2 exit sequence. A b_i/g_i flag value of zero corresponds to the L0 link state; the value of 1 corresponds to the L2 link state. The ATU-R may instead send an L2 Reject command indicating it is temporarily busy using reason code 01₁₆.

The ATU-R shall send a response command to an L2 request by the ATU-C within the time period defined in Table 7-17. An ATU-R shall not send an L2 Grant command if it has already sent an OLR request command and is awaiting a response.

9.4.1.7.4 L2 trim request by ATU-C

When sending the L2 Trim Request command, the ATU-C shall propose a new value of the PMD control parameter PCBds.

Upon receipt of the power management L2 Trim Request command, the ATU-R shall evaluate the parameter found in the L2 Trim Request message and the current operating conditions of the downstream receiver. If the parameters are invalid (i.e., not within the allowed encoding ranges), the ATU-R shall send a L2 Trim Reject command using reason code 02₁₆. If the parameters are valid but describe an operating condition that cannot be currently satisfied, the ATU-R shall send a L2 Reject command using reason code 04₁₆. If the parameters can be met, the ATU-R shall send an L2 Trim Grant command and follow procedures defined in 9.5.3.6. The L2 Trim Grant command shall describe the PCBds value that the ATU-C shall use to transmit a L2 exit sequence.

9.4.1.8 Clear eoc messages

The clear eoc command may be used by the G.997.1 function to transfer management octets from one ATU to another (see clause 6/G.997.1). The clear eoc command may be initiated by either ATU as shown in Table 9-24. The responses shall be using the command shown in Table 9-25. The clear eoc command shall consist of multiple octets. The first octet shall be clear eoc command designator shown in Table 9-3. The remaining octets shall be as shown in Table 9-24. The clear eoc response command shall be 2 octets. The first octet shall be the clear eoc command designator shown in Table 9-3. The second shall be as shown in Table 9-25. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-24/G.992.3 – Clear eoc commands transmitted by the initiating ATU

Message length (Octets)	Element name (Command)
variable	01 ₁₆ followed by the entire eoc message to be delivered at the far end All other octet values are reserved by the ITU-T.

Table 9-25/G.992.3 – Clear eoc command transmitted by the responding ATU

Message length (Octets)	Element name (Command)
2	80 ₁₆ ACK
3	81 ₁₆ NACK followed by: 1 octet for reason code All other octet values are reserved by the ITU-T.

Upon receipt of the clear eoc command, the ATU shall respond with an acknowledgement (ACK) message. The ATU shall deliver this message to the local G.997.1 management function. The message is delivered transparently. Whatever formatting was applied by the G.997.1 management function at the transmitting end is conveyed at the receiving end, e.g., block based format, variable length command format. The ATU may also reply with a NACK command with reason code Not Supported (value 04₁₆), indicating the clear eoc message cannot be delivered because the G.997.1 function does not support transport of physical layer OAM messages through the clear eoc (see clause 6/G.997.1).

9.4.1.9 Non-standard facility overhead commands

The non-standard facility (NSF) overhead command may be used to transfer vendor discretionary commands from one ATU to another. The NSF overhead command may be initiated by either ATU as shown in Table 9-26. The responses shall be using the command shown in Tables 9-26 and 9-27. The NSF overhead command shall consist of multiple octets. The first octet shall be NSF overhead command designator shown in Table 9-3 or Table 9-4. The command designator in Table 9-4 is for lower priority commands that should not interrupt the flow of normal priority commands in Table 9-3. The remaining octets of both standard and low priority messages shall be as shown in Table 9-26. The NSF overhead response command shall be 2 octets. The first octet shall be the NSF overhead command designator shown in Table 9-3. The second shall be as shown in Table 9-27. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-26/G.992.3 – Non-standard Facility (NSF) overhead commands transmitted by the initiating ATU

Message length (Octets)	Element name (Command)
variable	01 ₁₆ followed by: NSF identifier field NSF message field All other octet values are reserved by the ITU-T.

Table 9-27/G.992.3 – Non-standard Facility (NSF) overhead commands transmitted by the responding ATU

Message length (Octets)	Element name (Command)
2	Command 80 ₁₆ ACK
2	81 ₁₆ NACK All other octet values are reserved by the ITU-T.

Upon receipt of the NSF overhead command, the ATU shall respond with either an acknowledgement (ACK) message or a negative acknowledgement message (NACK). The ACK is used to indicate that the NSF identifier field is recognized. The NACK is used to indicate that the NSF identifier field or NSF message field is not recognized.

The combination of the NSF identifier field and NSF message field corresponds to a non-standard information block as defined in Figure 11/G.994.1, without the non-standard information length octet. The NSF identifier field consists of 6 octets. The first 2 octets are a country code as defined by ITU-T Rec. T.35. The remaining 4 octets is a provider code as specified by the country identified in ITU-T Rec. T.35. The NSF message field consists of M octets and contains vendor-specific information. The length and syntax of the NSF message field are not specified and are dependent upon the NSF identifier.

9.4.1.10 Test parameter messages

The PMD test parameters read commands shall be used to access the value of certain PMD test parameters maintained by the far ATU in accordance with the description of the PMD function. The local parameter values shall be retrieved as described in this subclause. The PMD test parameter read command may be initiated by either ATU as shown in Table 9-28. The responses shall be using the command shown in Table 9-29. The PMD test parameter read command shall consist of a two octets. The first octet shall be PMD test parameter command designator shown in Table 9-4. The second octet shall be one of the values shown in Table 9-28. The PMD test parameter read response command shall be multiple octets. The first octet shall be PMD test parameter read command designator shown in Table 9-4. The second shall correspond to received management counter read command. The remaining octets shall be as shown in Table 9-29. The octets shall be sent using the format described in 7.8.2.3 and using the protocol described in 7.8.2.4.

Table 9-28/G.992.3 – PMD test parameter read commands transmitted by the initiator

Message length (Octets)	Element name (Command)
3	01 ₁₆ Single Read followed by: 1 octet describing the test parameter id
3	02 ₁₆ Multiple Read Block followed by: 1 octet describing the subcarrier index
2	03 ₁₆ Next Multiple Read: All other octet values are reserved by the ITU-T.

Table 9-29/G.992.3 – PMD test parameter read command transmitted by the responder

Message length (Octets)	Element name (Command)
variable (see Note)	81 ₁₆ followed by octets for the test parameter arranged for the single read format
12	82 ₁₆ followed by octets for the test parameters arranged for the multiple read format
2	80 ₁₆ NACK All other octet values are reserved by the ITU-T.
NOTE – Variable length equals 2 plus length shown in Table 9-30.	

Upon receipt of one of the PMD test parameter read commands, the receiving ATU shall transmit the corresponding response message. If an unrecognised test parameter is requested, the response shall be a PMD test parameter command for NACK. The function of the receiving or transmitting ATUs is not otherwise affected.

The PMD test parameters are all derived according to the procedures in the PMD function sub-clause of this Recommendation. Following initialization, the PMD shall maintain training test parameters until the overhead command for update test parameters is received.

The parameters are transferred in the order and format defined in Table 9-30. During a test parameter read command for single read, all information for the test parameter is transferred. If the test parameter is an aggregate parameter, only one value is transferred. If the test parameter has a value per subcarrier, then all values are transferred from subcarrier index #0 to subcarrier index #NSC – 1 in a single message. The format of the octets is as described in PMD subclause. Values that are formatted as multiple octets shall be inserted in the response message most significant to least significant octet order.

During a test parameter read command for multiple read or next, information for all test parameters associated with a particular subcarrier as are transferred. Aggregate test parameters are not transferred with the PMD test parameter read command for multiple read or next. The subcarrier used for a PMD test parameter read command for multiple read shall be the subcarrier contained within the command. This subcarrier index shall be saved. Each subsequent PMD test parameter command for next shall increment and use the saved subcarrier index. If the subcarrier index reaches NSC, the response shall be a PMD test parameter command for NACK. The per subcarrier values are inserted into the message according to the numeric order of the octets designators show in Table 9-30. The format of the octets is as described in PMD subclause of this Recommendation. Values that are formatted as multiple octets shall be inserted in the response message most significant to least significant octet order.

Table 9-30/G.992.3 – PMD test parameter ID values

Test parameter ID	Test parameter name	Length for single read	Length for multiple read
01 ₁₆	Channel Transfer Function $Hlog(f)$ per subcarrier	$2 + NSC \times 2$ octets	4 octets
02 ₁₆	Reserved by ITU-T		
03 ₁₆	Quiet Line Noise PSD $QLN(f)$ per subcarrier	$2 + NSC$ octets	3 octets
04 ₁₆	Signal to noise ratio $SNR(f)$ per subcarrier	$2 + NSC$ octets	3 octets
05 ₁₆	Reserved by ITU-T		
21 ₁₆	Line Attenuation $LATN$	2 octets	N/a
22 ₁₆	Signal Attenuation $SATN$	2 octets	N/a
23 ₁₆	Signal-to-Noise Margin $SNRM$	2 octets	N/a
24 ₁₆	Attainable Net Data Rate $ATTNDR$	4 octets	N/a
25 ₁₆	Near-end Actual Aggregate Transmit Power $ACTATP$	2 octets	N/a
26 ₁₆	Far-end Actual Aggregate Transmit Power $ACTATP$	2 octets	N/a

In transferring the value of the channel transfer function $Hlog(f)$, the measurement time shall be inserted into the message, followed by the value m (see 8.12.3.1). The measurement time is included only once in a PMD test parameter response for single read. The measurement time is included in each response for multiple read or next multiple read.

In transferring the value of the quiet line noise $QLN(f)$, the measurement time shall be inserted into the message, followed by the n value (see 8.12.3.2). The measurement time is included only once in a PMD test parameter response for single read. The measurement time is included in each response for multiple read or next multiple read.

In transferring the value of the signal-to-noise ratio $SNR(f)$, the measurement time shall be inserted into the message, followed by the snr value (see 8.12.3.3). The measurement time is included only once in a PMD test parameter response for single read. The measurement time is included in each response for multiple read or next multiple read.

The values for test parameters defined with fewer bits than shown in Table 9-30, shall be inserted into the message using the least significant bits of the two octets. Unused more significant bits shall be set to 0 for unsigned quantities and to the value of the sign bit for signed quantities.

9.4.1.10.1 Single read command

Aggregate test parameters shall be retrieved using a single read and response procedure. Per subcarrier test parameters may be exchanged in a similar manner with a single read and response exchanged used to exchange all values for a test parameter, starting from subcarrier 0 to NSC .

9.4.1.10.2 Multiple read protocol with next

Per subcarrier exchange parameters may also be exchanged using shorter messages. The first command retrieves each test parameter for a requested subcarrier. A subsequent command retrieves all subcarrier test parameters for the next subcarrier. An invalid response is used to indicate a subcarrier index out of range or when the end of the subcarrier list has been reached.

Hargrave's Communications Dictionary

Frank Hargrave

HARGRAVE'S COMMUNICATIONS DICTIONARY

Frank Hargrave



**IEEE
PRESS**

The Institute of Electrical and Electronics Engineers, Inc., New York

This book and other books may be purchased at a discount from the publisher when ordered in bulk quantities. Contact:

IEEE Press Marketing
Attn: Special Sales
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Fax: +1 732 981 9334

For more information about IEEE Press products, visit the IEEE Online Store & Catalog: <http://www.ieee.org/store>.

© 2001 by the Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, 17th Floor, New York, NY 10016-5997.

All rights reserved. No part of this book may be reproduced in any form, nor may it be stored in a retrieval system or transmitted in any form, without written permission from the publisher.

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

ISBN 0-7803-6020-6
IEEE Order No. PC5869

Library of Congress Cataloging-in-Publication Data

Hargrave, Frank.

Hargrave's communications dictionary.

/ Frank Hargrave.

p. cm.

Includes index.

ISBN 0-7803-6020-6

1. Telecommunication—Dictionaries. I. Title.

TK5102.H37 2000

621.382'03—dc21

00-061416

first point of switching

first point of switching The first telephone company location at which routing occurs.

first window Of silica-based optical fibers, the transmission window at approximately 830 to 850 nm. See also *window*.

FIT An acronym for Failure In Time.

FITL An abbreviation of Fiber In The Loop.

FIX An acronym for Federal Information eXchange.

fixed access In personal communications service (PCS), access to a network in which there is a set relationship between a terminal and the access interface. A single *identifier* serves both the access interface and the terminal. If the terminal moves to another access interface, it assumes the identity of the new interface.

fixed priority-oriented demand assignment (FPODA) In networking, a medium access protocol in which stations must reserve slots on the network from a master station. These slots are allocated according to the stations' priority levels.

fixed routing A routing strategy in which packets or messages are transmitted between the source and destination over a well-defined and constant path.

fixed tolerance band compression A data compression method in which data values are stored or transmitted only when they fall outside a specified limit. The recipient of the data must assume that values are in the specified range unless an alert signal indicating otherwise occurs. As an example, assume the temperature at a particular point in a chemical plant is to be kept between 150°C and 180°C, a *fixed tolerance band compression* telemetry system would transmit temperature values only above 180°C or below 150°C.

flag (1) A two-state indicator/variable such as yes/no, true/false, on/off, error/no-error, and ready/not-ready. It is a variable set at one point in a program for use later in the same or different programs. It informs these programs that a condition has been met or not met. (2) In synchronous transmission systems, the *flag* is a specific bit pattern that marks the beginning and end of a packet (frame). See also *flag character*. (3) A deprecated term for a mark (stop) signaling element.

flag character In X.25 packet switching technology, a special character (0111 1110) that is included at the beginning and end of every link access protocol-balanced (LAP-B) frame to indicate a frame boundary. The protocol uses *bit stuffing* to ensure that this sequence never occurs anywhere else in the packet. *Flag characters* are used in bit-oriented protocols, such as Advanced Data Communication Control Procedures (ADCCP), Synchronous Data Link Control (SDLC), and High Level Data Link Control (HDLC). Also called a *flag sequence*.

flame On bulletin board systems, a vitriolic criticism or insult of a *person* for his or her ideas, beliefs, postings; just because of a disagreement with those postings, ideas, and so on. Some network users tend to state their opinion in an inflammatory manner that they would never use were it not for the anonymity of a network connection. *Flaming* is not a respected pastime. DON'T do it.

flame bait Any network posting that is virtually guaranteed to draw a plethora of flame responses; e.g., any type of advertisement posted to a newsgroup.

flame war A sustained exchange of flames between two (or more) network users. Generally, they start as disputes as to the "proper" resolution of some issue and degrade into messages bent on undermining the opponent (ignoring the original issue).

214

flip-flop

flash A quick depression and release of the hook switch on a telephone instrument. The *flash* is a signal to the PBX or Centrex that a system control command will follow, e.g., a command to transfer the call to a different extension. Also called a *hook-flash* or *hook-switch flash*.

flash cut The "instantaneous" transfer from one switching system to another. For example, when installing a new PBX, the old system is removed and the new one is installed; the user calls out using the old system on Friday, and Monday the new system must be used.

flash memory A type of nonvolatile memory that contains both RAM and EEPROM memory. During normal operation, data are written to and read from the RAM portion; during power-off times, data are stored in the EEPROM. Before power is removed from the RAM portion of the chip, a signal tells the chip to copy relevant portions of the RAM to the EEPROM. When power is restored, these data are restored to the proper locations in RAM.

Flash memory is an attempt to solve two shortcomings of EEPROM memory, primarily the inherent slow writing speed and secondarily the relatively limited number of times to which EEPROM memory can be written. Also called *flash ROM*.

flat fading Fading in which all frequency components of a received radio signal vary in the same proportion simultaneously.

flat rate service Telephone service in which a single payment permits an unlimited number of local calls to be made without further charge for a specified period of time.

flat structure A strategy in which each entity is unique and in which there is no logical, physical, or other relationship between them. This strategy may be used for files names or network node names. Accessibility to such a structure is through a lookup table. See also *hierarchical name structure*.

flat weighting In a noise-measuring set, a measurement based on an amplitude-frequency characteristic that is flat over a stated frequency range. Noise power is expressed in dBm($f_1 - f_2$) or dBm($f_1 - f_2$).

In telephony, the first frequency (f_1) is frequently omitted and assumed to be 30 Hz. For example: a "3 kHz flat weighting" assumes a flat frequency response between 30 Hz and 3 kHz, and a "15 kHz flat weighting" assumes a flat frequency response between 30 Hz and 15 kHz.

flavor Slang for a different version of software or hardware. For example, Berkeley Software Distribution (BSD) and system V are two *flavors* of UNIX.

Fleming's rule A mnemonic rule stating that if the thumb of the right hand points in the direction of current flow, then the curled fingers point in the direction of the magnetic field that encircles the current. Further, if the curled fingers of the right hand point in the direction of current flow in a solenoid, then the thumb points in the direction of the magnetic field inside the solenoid. Also called the *right-hand rule*.

flexible release The ability of a switching system to release an established circuit when either station goes on-hook. Also called *either party control*.

flexible routing The ability to choose different physical paths through a network for different calls to the same station as the circumstances warrant.

FLIH An abbreviation of First Level Interrupt Handler.

flip-flop A circuit or device that may assume one of two alternative stable states. During the transition from one stable state to the other, the output state is unstable, i.e., for the very short period during

OVER 600,000 SOLD

#1 Best
Seller

NEWTON'S TELECOM DICTIONARY

Covering Telecommunications,
Networking, Information Technology,
Computing and the Internet

MORE THAN 21,000 TERMS DEFINED

LIBRARY COPY

© 2003

19th

Updated, Improved and Expanded Edition
by Harry Newton

CMPBooks

NEWTON's TELECOM DICTIONARY

copyright © 2003 Harry Newton

email: Harry@HarryNewton.com

personal web site: www.HarryNewton.com

business web site: www.TechnologyInvestor.com

All rights reserved under International and Pan-American Copyright conventions,
including the right to reproduce this book or portions thereof in any form whatsoever.

Published by CMP Books

An imprint of CMP Media LLC

Main office: CMP Books, 600 Harrison St., San Francisco, CA 94107 USA

Phone: 415-947-6615; Fax: 415-947-6015

Sales office: CMP Books, 12 West 21 Street, New York, NY 10010

Phone 917-305-3333; Fax 212-206-0387

www.cmpbooks.com

Email: books@cmp.com



CMP

United Business Media

For individual orders, and for information on special discounts for quantity orders,
please contact:

CMP Books Distribution Center, 6600 Silacci Way, Gilroy, CA 95020

Tel: 1-800-500-6875 or 408-848-3854; Fax: 408-848-5784

Email: cmp@rushorder.com; Web: www.cmpbooks.com

Distributed to the book trade in the U.S. by:

Publishers Group West, 1700 Fourth Street, Berkeley, California 94710

Distributed in Canada by:

Jaguar Book Group, 100 Armstrong Avenue, Georgetown, Ontario M6K 3E7 Canada

Printed in the United States of America

This book is also sold through www.Amazon.com, www.Fatbrain.com and
www.BarnesAndNoble.com

Distributed to the book trade in the U.S. and Canada by Publishers Group West
1700 Fourth St., Berkeley, CA 94710

Fax: 408-848-5784

cmp@rushorder.com

ISBN Number 1-57820-307-4

March 2003

Nineteenth Edition

Matt Kelsey, Publisher

Ray Horak, Senior Contributing Editor

Saul Roldan, Cover Artist

Lisa Giaquinto, Project Manager

Brad Greene, Text Layout

Fish Job

2. Fish food refers to sales leads among a group of salespeople. Basically sales people throw a lead out, like fishfood and all the others come up to see if they can sell it. Contributed by Dino Guglietta.

Fish Job Running cables inside walls. Usage: "That fish job is too tough for a rookie."

Fish Tape Non-conductive tape with a reinforced fiberglass core and slippery outer nylon coating which slides easily through conduit without jamming. The idea is to push the tape through, attach it to the cable and pull the cable back. You also might use wire pulling lubricants to make the job even easier. They come in various formulations — for use in different temperatures, for pulling different cable, etc.

Fishing Expedition An investigation that has no clearly defined objective. The goal is to find something immoral, incriminating, unethical or illegal.

FISK Fax a dISK. Method of sending information on 3 1/2" disks painlessly across phone lines. Plug your disk into a fisk machine, choose which files you want to send, dial the number you want to send your files to and walk away.

FITC Fiber To The Curb. See FTTP.

FITH Fiber To The Home. (I kid you not. That's what it stands for.) See FTTP.

FITL Fiber In The Loop. An outside plant architecture, deployed by some telephone companies for providing broadband services to subscribers. In the FITL architecture, SNET fiber runs from the telephone company central office to an optical networking unit (ONU) at the curb, which serves 8064 subscribers. Subscribers are served from the ONU in a star topology, each home with a drop of coax, twisted pair, or composite coax-and-twisted-pair. With FITL, an individual video or voice signal is switched to each subscriber. Therein lies the major difference between FITL and the alternative broadband architecture, a hybrid fiber/coax bus. In the bus architecture, all signals serving dozens of subscribers are multiplexed and sent to all subscribers. See also Fiber Optics, FTTC, FTTH, Local Loop and POP.

Five by Five The term 5x5 describes the transmit and receive quality of teletype communications (on a scale of 1 to 5, with 5 being the best) between two stations. Example, 1x5 means transmitting poorly but receiving excellent. The source of this info comes from Alan DiGilio's training, as a former Naval Flight Officer, in the use of cryptographic radio teletype equipment. ("craii" as we called it.) Amongst my comrades in arms, when we asked each other how transmissions went, we would slip into slang by saying "Fivers," to mean 5x5.

Five Nines 99.999%. Five nines typically refers to the reliability of a system (computer, telephone system, etc.) that works 99.999% of the time. This is far better than the present common standard of three nines, i.e. 99.9%.

Fivers See Five by Five.

FIX Federal Internet Exchange. A connection point largely serving to interconnect network traffic from MILNET (MILitary NETWORK), NASA Science Net and other federal government networks, as well as providing those network users with access to the Internet. FIX-EAST is located at the University of Maryland in College Park, Maryland; FIX-WEST is located at the NASA Ames Research Center at Moffet Field between Sunnyvale and Mountain View, California. See also CIX, MAE and NAP.

Fixed Attached and permanent. The opposite of mobile. See Fixed Wireless.

Fixed Condenser A condenser, the plates of which are stationary and the capacity of which cannot be changed.

Fixed Disk Old name for a hard disk.

Fixed End System F-ES. A non-mobile end system. A host system that supports or provides access to data and applications.

Fixed Format A way of communicating in which everything to be sent follows a predetermined sequence, i.e. it fits into a specific length and format. The idea is to allow you to predict message length, the location of the message, where the control characters are, etc.

Fixed IP Address See IP Addressing.

Fixed Length Records A set of data records all having the same number of characters in them. Think of a database of name, address, city, state, zip. Clearly, not everyone's record will be the same length. In order to make a fixed length record, the computer will pad the record with "padding characters" which the computer will ignore when it reads the record. But by including the padding characters it has effectively given everyone the same fixed length record.

Fixed Line A wired phone line. The one you have in your house, as compared to your wireless phone line, the one you have in your pocket.

Fixed Link A communications link between two fixed points. Such links may be unidirectional (e.g. carrying television program material to a transmitter) or bi-directional (e.g. carrying telephone traffic), and may be point-to-point or point-to-multipoint.

Fixed Loop A services feature available in some switching systems that permits an attendant on an assisted call to retain connection through the attendant position for the duration of the call. The attendant will normally receive a disconnect signal when the call has been completed.

Fixed Priority-Oriented-Demand Assignment FPODA. Medium access technique in which one station acts as master and controls channel based on requests from stations.

Fixed Rate A fixed monthly price. See also Flat Rate.

Fixed Satellite Service A radiocommunication service between Earth stations as specified fixed points when one or more satellites are used; in some cases this service includes satellite-to-satellite links, which may also be effected in the inter-satellite service, the fixed-satellite service may also include feeder linker for other space radiocommunication services.

Fixed Satellite System FSS. A system of Geosynchronous Earth Orbiting (GEO) satellites. GEOs are positioned in equatorial orbits approximately 22,300 miles above the Earth's surface. Positioned in this manner, they are synchronized with the rotation of the Earth. Therefore, they are always (more or less) fixed in the same physical location relative to the Earth's surface. This allows satellite antennas on the ground to be fixed and not have to move to follow the movement of the satellites. Actually, fixed orbit satellites do slide out of their orbit slightly. Their onboard rocket engines are then started and they are brought back into geosynchronous orbit. See also GEO. Contrast with LEO and MEO.

Fixed Wireless See Fixed Wireless Local Loop.

Fixed Wireless Local Loop FWLL. Imagine a community of 100 people spread out in a huge area in one of the Western states in the United States — e.g. Montana or Wyoming. Imagine a city of people eager to get faster Internet access and better, cheaper phone service than their local phone company can provide. The local loop is best described as the "last mile" of phone service. It's the distance between you, the customer, and the switching office down the street or across the county that's owned by the local telephone company. In most cases, local loop service uses old-fashioned twisted copper wire installed and provided by the ILEC — the incumbent local exchange company (your local phone company). However, several phone companies and several of their competitors are installing coaxial cable, fiber optics, their own cable and now fixed wireless, also called "Wireless Fiber." Such systems operate at the 38 GHz portion of the spectrum. They generally consist of a pair of digital radio transmitters placed on rooftops — one at one end at the central office and the other end at the customers' offices. It's called "fixed" to contrast it with "mobile," e.g. cellular. In order to attract customers, some fixed wireless providers are offering higher data transmission rates than wire. In 1998, for instance, one firm announced a new, fixed wireless network that would carry 128 Kbps of digital transmission right into most households. Based on the 10 MHz spectrum, the new system would connect a home to a digital switching center via a neighborhood antenna mounted on a utility pole or other structure. A single antenna would serve up to 2,000 homes. Meanwhile, the customer would only need to secure a transceiver to the side of their house.

FL Fault Locating.

FL-port Fibre Channel term. Port that connects an FC-AL to a fabric.

Flag 1. A variable in a program to inform the program later on that a condition has been met.

2. In synchronous transmission, a flag is a specific bit pattern (usually 01111110) used to mark the beginning and end of a "frame" of data. Frame Relay and lots of other protocol use this approach in order to delineate one frame from another, and to allow the devices in the network to synchronize on the rate of transmission for purposes of improved bandwidth efficiency. See Frame Relay, Synchronous and Zero Stuffing.

3. Fiberoptic Link Around The Globe. The longest man-made structure in the world, Flag is a 17,000 mile long fiber optic cable made of four strands of fiber, carrying 10 gigabits of data per second. The cable is laid mostly undersea, linking Japan to Britain, making land in China, Thailand, the United Arab Emirates, Italy and six other countries. Companies in most of those countries own a share of Flag's bandwidth, however the United States Verizon is the biggest owner by far with a 38% share, which it leases out. Each strand of Flag's fiber is unidirectional, i.e. you need two to make a conversation, one for going and one for coming.

Flag Fall Also spelled flagfall. Older taxis had a metering system which had a large metal lever facing vertically up. It was called a flag because it sort of looked like one. When a customer got into the taxi, the driver pulled the metal flag down. This action started the meter. All taxis typically charge a fixed money the moment you get into the taxi and then

Fundamentals of DSL Technology

Edited by
Philip Golden ■ Hervé Dedieu ■ Krista S. Jacobsen

Fundamentals of DSL Technology

**Edited by
Philip Golden
Hervé Dedieu
Krista Jacobsen**



Auerbach Publications

Taylor & Francis Group
Boca Raton New York

Cover designer: Chris Massey

Published in 2006 by
Auerbach Publications
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2006 by Taylor & Francis Group, LLC
Auerbach is an imprint of Taylor & Francis Group

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 0-8493-1913-7 (Hardcover)
International Standard Book Number-13: 978-0-8493-1913-6 (Hardcover)
Library of Congress Card Number 2004062330

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

No part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC) 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Fundamentals of DSL technology / editors, Philip Golden, Herve Dedieu, Krista Jacobsen.

p. cm.

Includes bibliographical references and index.

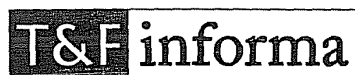
ISBN 0-8493-1913-7 (alk. paper)

I. Digital subscriber lines. I. Golden, Philip. II. Dedieu, Hervé. III. Jacobsen, Krista.

TK5103.78.F86 2005

621.382--dc22

2004062330



Taylor & Francis Group
is the Academic Division of T&F Informa plc.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the Auerbach Publications Web site at
<http://www.auerbach-publications.com>

9.4 Interleaving

As mentioned in the abstract of Chapter 1, one of the key reasons for the wide deployment of DSL is that it runs over existing telephone lines. Not having been designed for high-speed digital communications, these existing lines are prone to interference from external sources. This interference causes what is normally called impulse noise in DSL recommendations. Impulse noise can be caused by telephone ringing (see Section 1.6), picking up the telephone (see Section 1.7), dial pulse (see Section 1.9) [Brown 1999], or any number of other external factors such as weather, intermittent radio frequency interference (see Section 3.2), or even large appliances.

Clearly it is difficult to characterize the exact characteristics, frequency, or duration of impulse noise. As a guide, however, two “characteristic” noise impulses are captured in the ADSL standard, T1.413 Issue 2 [T1.413 1998]. These impulses last for about 22 μ s. However, impulse noise can typically be 10–20 times longer or more. Interleaving is used with Reed–Solomon coding to correct bursts of errors caused by noise of this sort.

An interleaver is a device that accepts codewords from a finite alphabet and returns the identical codewords but in a different order. Combined with Reed–Solomon coding, an interleaver can spread long strings of errors over several codewords. The spreading is sufficient to correct a burst of errors if the burst causes no more than t errors in any one codeword in a t -error correcting Reed–Solomon code (or s erasures in a s -erasure correcting Reed–Solomon code if erasures are used). Long bursts of errors can result from impulse noise. But errors can also result from the concatenated coding where the first stage of decoding generates bursts of errors as in trellis decoding using a Viterbi detector.

The simplest interleaver to consider is a block interleaver. With a block interleaver, codewords are written into a rectangular array in columns and are read out in rows. There are I rows and d columns where I is the codeword size (or a divisor of the codeword size) and d is the interleaver depth. A block interleaver with $I = 7$ and $d = 4$ is shown in Figure 9.12. An example of eight consecutive errors is also shown in the figure. In this case, if this code could correct up to $t = 2$ errors and the codeword size, n is equal to I , the burst of eight errors would be corrected.

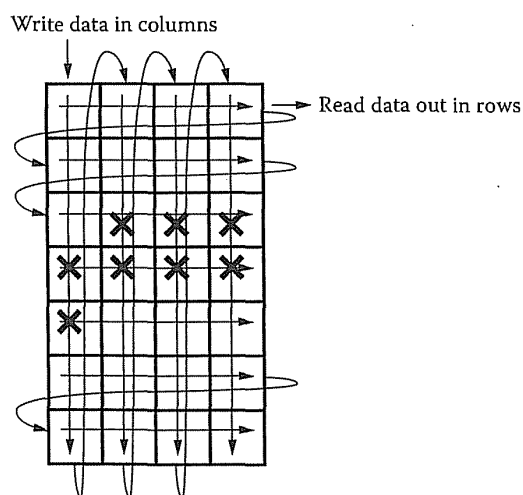


FIGURE 9.12

Block interleaver with seven rows and four columns. Codewords are written in columns and then read out in rows. As an example, each of eight consecutive errors is marked by an X. Each codeword has only two errors.

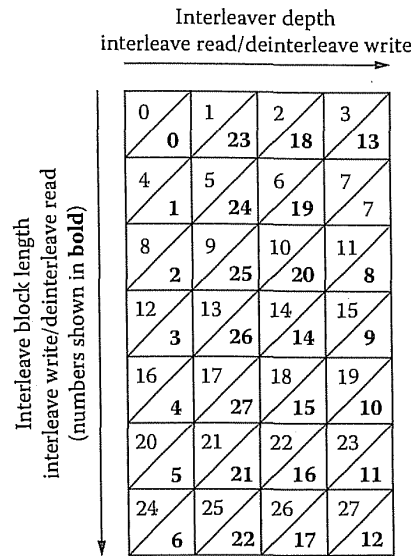


FIGURE 9.13

Convolutional interleaver with read and write addresses shown.

Because block interleavers do not lend themselves to low-memory, efficient implementations, many DSL systems use convolutional interleaving introduced originally by Ramsey [Ramsey 1970] and Forney [Forney 1971]. In general, a convolutional interleaver imposes a different delay on each input symbol. If i denotes the symbol index within a group of I symbols so that $i = 0, 1, \dots, I - 1$, symbol i experiences a delay of $i \cdot (d - 1)$ with d the interleaver depth. The deinterleaver performs the inverse operation delaying symbol i by $(I - i - 1)(d - 1)$. The overall delay of the interleaver/deinterleaver pair is $(I - 1)(d - 1)$.

A convolutional interleaver is shown in Figure 9.13 for $I = 7$ and $d = 4$. As with the block interleaver, symbols are written into the interleaver in columns and read out in rows. However, unlike the block interleaver, it is not necessary to wait to fill the entire block before reading. As shown in Figure 9.13, the first symbol, symbol 0, is written then read immediately with no delay. Symbol 1 is written, then read at time $k = 4$ delayed by 3 samples. Symbol 2 is written then read out at time 8 delayed by 6 samples. For this example, the ordering of the input symbols in the output sequence is

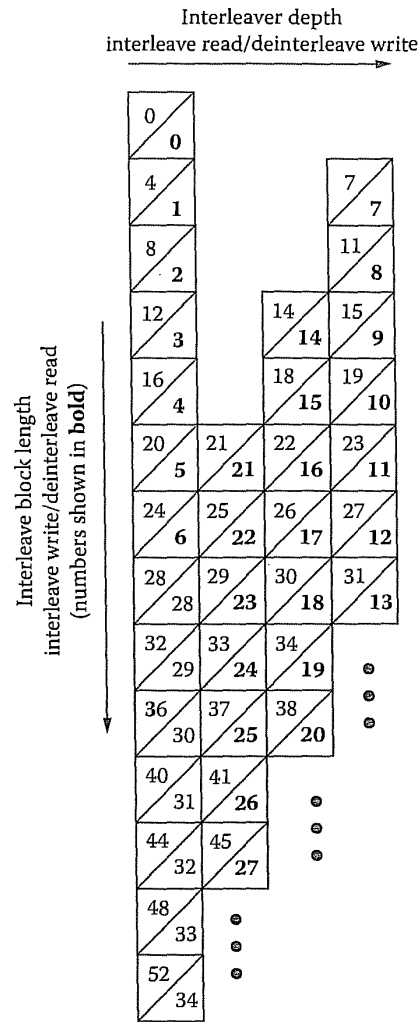
$$0, 4, 8, 12, 16, 20, 24, 7, 11, 15, 19, 23, 27, 31, 14, 18, \dots$$

In general, for a convolutional interleaver, the ordering of the input symbols in the output sequence will be

$$0, d, 2d, \dots, (I - 1) \cdot d, I, I + d, I + 2d, \dots$$

In order to ensure that an input is never repeated in the output, as required in a valid interleaver, the block length I and the depth d must be co-prime, meaning that they share no common factors aside from 1.

When I and d are not co-prime but $I + 1$ and d are co-prime, it is possible to add a dummy row to the interleaver to give it $(I + 1)$ rows instead of I rows. In this case, no input data is written into or read from the $I + 1$ st row of the interleaver. The dummy row is added only to create a valid interleaver. In the ADSL and ADSL2 recommendations, the interleaver depth is restricted to be a power of 2 from 2^0 to 2^6 . In this case, only odd values are co-prime with d . In ADSL and ADSL2, I is set to the Reed–Solomon codeword size, n , and a dummy row is always added whenever n is even.

**FIGURE 9.14**

Cascading column representation of a convolutional interleaver with $d = 4$ and $I = 7$.

As shown in Figure 9.13, a convolutional interleaver is very similar to a block interleaver except that the blocks of I symbols are staggered. Another way to see the convolutional interleaver is as in Figure 9.14. In this view of a convolutional interleaver, blocks of I symbols are added in a constantly descending pattern; as if playing a game of Tetris with the symbol blocks coming from the bottom. In this analogy, the new block is always added to the column farthest from the bottom or to the column farthest to the left if columns share the same height. This means that rows in the interleaver do not appear in the same position in adjacent symbol blocks. However, the error-spreading properties of the block interleaver are preserved in a convolutional interleaver.

Interleavers are characterized by their delay, memory requirements, and spreading [Heegard 1999]. The delay of the interleaver is the total delay between the interleaver input and the deinterleaver output caused by the interleaver/deinterleaver. With a block interleaver, a straightforward implementation will wait for the block of $I \times d$ to fill before starting to read out the codewords. Therefore, the interleaver and de-interleaver delay combined is $2 \times I \times d$. The desire to correct long error bursts is often at odds with the requirement for small delay in DSL systems. In applications such as video conferencing, voice, and gaming,

small delay is critical. Even in general Internet use, small delays yield higher throughput. In general, it is desirable to minimize the delay for a given immunity to impulse noise.

Interleaving is typically one of the largest consumers of memory on a DSL transceiver chip. Therefore, it is critical to use the smallest possible amount of interleaver memory. The smallest amount of memory required to build an interleaver/deinterleaver pair is equal to the total delay of the interleaver/deinterleaver [Heegard 1999]. Typically, for memory-optimized interleavers, the interleaver and deinterleaver memory size is nearly the same. As mentioned, the delay of a convolutional interleaver is $(I - 1)(d - 1)$. As will be discussed in more detail later, it is possible to design a convolutional interleaver with very nearly the theoretical minimum amount of memory equal to $(I - 1)(d - 1)$ symbols or $(I - 1)(d - 1)/2$ symbols for either the interleaver or deinterleaver alone.

Regarding spreading, the output of an (I, d) convolutional interleaver can be described as re-ordering the input sequence so that no contiguous sequence of n_2 symbols in the re-ordered sequence contains any symbols that were separated by fewer than d symbols in the original ordering [Ramsey 1970]. The value of n_2 depends on the interleaver parameters and is

$$n_2 = I - \left\lceil \frac{I}{d} \right\rceil, \quad (9.27)$$

where $\lceil \cdot \rceil$ means “greatest integer greater than.”

PROOF As mentioned earlier, the ordering of the input symbols in the output sequence is

$$0, d, 2d, \dots, (I - 1) \cdot d, I, I + d, I + 2d, \dots$$

In the first I symbols and all groups of I symbols thereafter, the distance between adjacent symbols is always d . The distance between symbol I and symbol $0 \leq k < I$ is $|I - k \cdot d|$. The objective is to find the smallest k such that $|I - i \cdot d| \geq d$ for $i = k \dots I - 1$. In order for $|I - i \cdot d| \geq d$ for $i = k \dots I - 1$, I must either be always greater than $i \cdot d$ or always less than $i \cdot d$ because if it were not, there would have to exist some value of i for which $|I - i \cdot d| < d$. If I is always greater than $i \cdot d$, this would mean that $I > (I - 1) \cdot d$, which can only happen if $I = 1$ or $d = 1$. Neither of these cases is of interest, and therefore I must be less than $i \cdot d$ for $i = k \dots I - 1$ and must therefore be less than $k \cdot d$. The smallest k such that $I < k \cdot d$ can be found as $k = \lceil \frac{I}{d} \rceil$. Symbol k is separated from symbol I by $I - k$ symbols and $I - \lceil \frac{I}{d} \rceil$ is therefore the maximum value of n_2 . ■

Effectively, for DSL implementations, interleaver spreading amounts to error correction capabilities. For all of the interleaving methods considered here, a t -error (s -erasure) correcting Reed–Solomon code plus interleaver with depth d can correct a burst of up to $t \cdot d \cdot I/n$ or $(s \cdot d \cdot I/n)$ symbols. Bursts must be separated sufficiently so that two or more bursts do not corrupt the same codeword. As long as n/I is an integer, if bursts are periodic, the period must not exceed $n \cdot d$ symbols.

9.4.1 Optimum Memory Implementation Using Tong’s Method

As mentioned earlier, a convolutional interleaver/deinterleaver pair can be implemented with $(I - 1)(d - 1)$ memory symbols. In [Ramsey 1970] a method for doing this is outlined using shift registers. Because interleaver memory represents such a substantial cost in DSL systems and because shift registers are very inefficient, this method is not suitable for implementation. Random access memory (RAM) is used instead. It is possible to implement a convolutional interleaver using a single RAM using the addressing as shown in Figure 9.13. However, this scheme uses more than twice the minimum amount of memory requiring

TABLE 9.6

Example of Tong's Addressing Method with $I = 4$, $d = 3$

Symbol	0	1	2	3
Delay	1	3	5	7
Address	0	0	1	2
	0	0	3	1
	0	0	2	3
	0	0	1	2
	\vdots	\vdots	\vdots	\vdots
Period	1	1	3	3

$2 \cdot I \cdot d$ memory symbols for the complete interleaver/deinterleaver pair. A near-optimal implementation of a convolutional interleaver is described by Tong [Tong 1998]. In Tong's method, the same memory address is used for both reading and writing. For each address, data is read and then written. With this method, Tong recognizes that if a symbol, i , is delayed by $(i\%I) \cdot (d - 1)$ symbols, where $i\%I$ means "the integer remainder of i/I ," this must mean that the memory location in which symbol i was written will appear again $(i\%I) \cdot (d - 1)$ symbols later. By reading and then writing the same memory location each cycle, this method introduces a fixed delay of 1 additional symbol in the interleaver and 1 additional symbol in the deinterleaver so that the total delay is actually $(I - 1)(d - 1) + 2$.

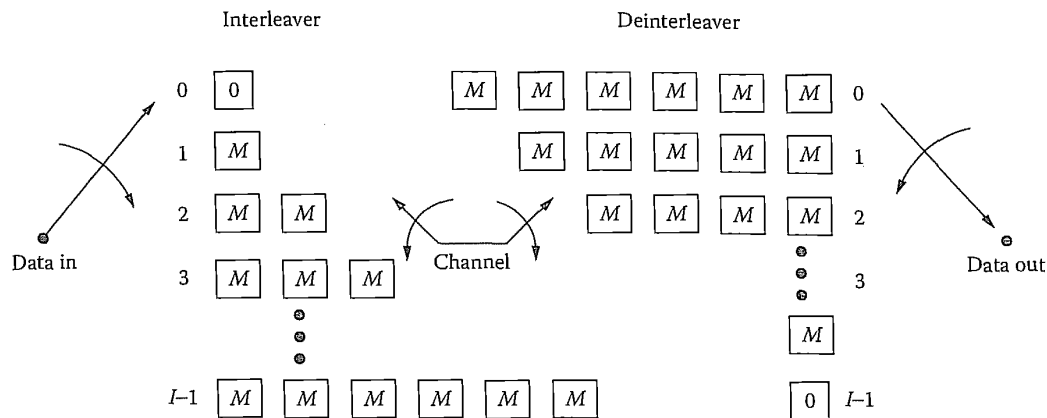
As an example, the memory addresses are shown in Table 9.6 for the case $I = 4$ and $d = 3$. In this case, $i\%I$ is 0, 1, 2, and 3 and continues to repeat as index i increments. The delay of symbol i is shown as $i \cdot (d - 1) + 1$ for $i = 0 \dots 3$. The process begins with address location 0 and then repeats address location 0, 1 symbol later because the delay is 1, then 3 symbols later because the delay is 3, and so on. Then address 1 occurs in symbol 2. Because this symbol has a delay of 5, five spaces are counted before address 1 appears again. The process of moving across the rows of the table and filling in new addresses if necessary continues until the entire sequence eventually repeats. In this case, the sequence repeats after 12 symbols. Along the first and second columns, the address sequence is 0, 0, 0, \dots and in the third and fourth columns the sequence is 3, 2, 1, 3, 2, 1, \dots but offset from each other.

In general, there will be I different address sequences, one for each delay value, which can always be written as decrementing, contiguous counters. Each counter can have a different start, maximum, and minimum value. In this example, the start, maximum, and minimum addresses for the first two delay values (1 and 3) are all zero. The start addresses for the last two delay values are 1 and 2, respectively, with the maximum and minimum addresses for both of these last two delays being 3 and 1, respectively. In Tong's method, three arrays of length I are stored, A , L , and U , with the current address, the lower limit on the address, and the upper limit on the address, respectively. The algorithm continues to cycle through each of the I columns, and at column i , the address used to read and write is $A(i - 1) - 1$ if $A(i) > L(i)$ or $U(i)$ otherwise.

The deinterleaver can work exactly the same way. With the deinterleaver, the delays are $(I - 1)(d - 1) + 1$, $(I - 2)(d - 1) + 1$, \dots , 1. The memory required for either the interleaver or deinterleaver is $(I - 1)(d - 1)/2 + 1$ plus the memory required to store A , L , and U which is typically much smaller than the interleaver or deinterleaver memory itself.

9.4.2 Forney's Triangular Interleaver

For convolutional interleavers, the only restriction on I and d is that they be co-prime. And even if they are not co-prime, it is usually true that $I + 1$ and d are co-prime and the interleaver can be implemented with a dummy row.

**FIGURE 9.15**

Forney's triangular interleaver implemented as a series of shift registers.

However, Forney [Forney 1972] and Berlekamp and Tong [Berlekamp 1985] recognized that the implementation of a memory-optimized interleaver can be simplified by constraining the relationship between I and d . In [Berlekamp 1985], Berlekamp and Tong demonstrate a helical interleaver where the interleaver depth is either $I - 1$ or $I + 1$. In [Forney 1972], Forney presents what is often referred to as a triangular interleaver named for its shape when implemented as a series of shift registers as shown in Figure 9.15. In the triangular interleaver, $d = M \cdot I + 1$ where M is any positive integer. A triangular interleaver is used in both DMT and QAM VDSL1. For DSL applications at least, the triangular interleaver is usually considered flexible enough to meet the performance requirements.

As shown in Figure 9.15, the triangular interleaver can be seen as a series of shift registers. Each box labelled with "M" represents M storage locations. The box labelled "0" means no storage; data written into a 0 storage block comes out immediately. Data is written into the interleaver on the left one row at a time in a cyclic pattern. As data is written, delayed data is shifted out the right so for every input symbol there is a delayed output symbol.

The example in Figure 9.14 showed that the columns of I symbols are shifted downward. In Figure 9.14, the fourth column is shifted down one row, then the third and second columns are shifted an additional two rows each. The triangular interleaver imposes a regular pattern on the downward shifts as shown in Figure 9.16. The interleaver in Figure 9.16 has $I = 4$, $d = 9$, and $M = 2$. A similar type of regular pattern is formed with the helical interleaver.

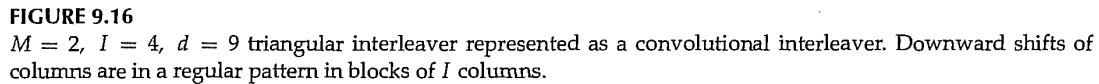
Because of the regular pattern, it is not necessary (but still possible) to use Tong's method to implement these interleavers using a single memory. As an example, an interleaver with $n = 4$ and $I = 5$ is shown in Table 9.7. Every "orbit" or column in the table has the same period. This will always be true with the helical or triangular interleaver. Furthermore, the address sequence can always be derived with the following algorithm assuming that y is the interleaver address and the total memory size is $\frac{(I-1)(d-1)}{2} + 1$:

```

y = 0      // starting address
while (there is more data to process)
  for i = 0 to I - 1
    y = (y + i * M) mod (total memory size)
  end
end

```

For the deinterleaver, i , the algorithm counts down from $I - 1$ to 0 instead of up from 0 to $I - 1$.



The advantage of the triangular or helical interleaver is the relatively simple implementation, with respect to Tong’s method, of a memory optimal interleaver. However, this comes at the price of reduced flexibility. When the size of interleaver memory is not a primary concern—either because the interleaver memory is off-chip or the total amount of interleaver memory does not significantly influence the overall complexity—then it is simple and sufficient to implement a convolutional interleaver using twice the optimal amount of memory. As DSL speeds increase, the interleaver memory size requirement increases and

Addressing for Forney's Triangular Interleaver Using Tong's Method. $I = 4, d = 5$

Symbol	0	1	2	3
Delay	1	5	9	13
Address	0	0	1	3
	6	6	0	2
	5	5	6	1
	4	4	5	0
	3	3	4	6
	2	2	3	5
	1	1	2	4
	0	0	1	3
	6	6	0	2
	:	:	:	:
period	7	7	7	7

it becomes more difficult to implement the interleaver with off-chip memory. Therefore, in VDSL, having a memory optimal interleaver implementation is generally required.

9.4.3 Error Correction Comparison of DMT and Single-Carrier Modulation

As line code battles continue, debates rage on the relative merits of DMT versus single-carrier in combating impulse noise. It is fair to say that there is a time/frequency duality between the two line codes. Because DMT contains many narrow frequency subchannels, a narrowband noise impulse will corrupt only a few DMT subchannels, whereas the same impulse could corrupt all single-carrier symbols for its duration. However, a very short, wideband impulse will corrupt only a few single-carrier symbols, whereas, in the worst case, if it lies on the border between two DMT frames, it could corrupt two entire DMT frames, which is about 500 μ s worth of data in all current DMT standards.

Interestingly, both the QAM and DMT VDSL1 specifications mandate 500 μ s of impulse noise protection using erasures. Correcting 500 μ s of data in either line code requires the same amount of memory at any given data rate. Therefore, those who drafted the competing line code proposals saw no particular advantage in terms of burst error correction for either line code.

9.4.4 Erasures

Recall from Section 9.2.5 that a Reed–Solomon code with r redundant symbols can be used to correct up to $2t + s \leq r$ symbols where s is the number of errors in known locations (erasures) and t is the number of errors in unknown locations. Because of the substantial memory requirements of interleavers and the desire to minimize the delay, it is highly desirable to take advantage of the factor of two improvement in error correction offered by using erasures. However, determining the location of errors is not trivial.

One way to try to determine the location of errors is to look for spikes in the error between the constellation points and the received, equalized input signal. Typically, these errors are computed for other purposes such as equalizer updates and trellis decoding. However, the difficulty with this method is that if erasures are flagged too frequently, the random error correcting capability of the code can be compromised. If flagged too infrequently, errors will be missed. A method that takes advantage of the inner trellis code in ADSL and ADSL2 to avoid some of the problems with too many or too few erasures is described in [Toumpakaris 2003a].

However, the VDSL1 recommendations do not use inner trellis codes, and ADSL connections are not required to use an inner trellis code.⁵ Therefore, an alternate method is desired. A relatively simple method that takes advantage of the structure of convolutional interleaving is described in [Berlekamp 1985]. A very similar method is also described in [Toumpakaris 2003b]. In these methods, erasures are predicted based on detected errors in previous codewords.

Because of the cascading nature of the columns in the convolutional interleaver as shown, for example, in Figures 9.14 and 9.16, a burst of errors will always begin at the bottom of one of the columns and, therefore, at the end of a codeword. Therefore, even for a very long burst, there will be only be a few errors in the first few corrupted codewords and they can be detected and corrected without erasures. However, if more than one error is detected toward the end of a codeword, or if errors are detected in similar positions across more than one codeword, it can be assumed an error burst has started. The error positions can continue to be tracked to determine whether the burst continues.

⁵ ADSL2 requires support for trellis codes, but the receiver can decide to disable it for any reason.

This method is simpler in the case of the triangular or helical interleavers because of the regular structure. However, it is still possible even with general convolutional interleaving.

Although this method is designed for impulse noise, it is also possible that the technique will be beneficial when considering the concatenated codes that are described in Section 9.5. With concatenated codes, errors that slip through the inner code often occur in bursts. It is possible that erasures can be used to maintain high coding gains using smaller interleaving depths than would be required without erasures. The author is not aware of any work that has been done on this topic to date.

9.5 Concatenated Coding

Concatenated coding was introduced by Forney in his doctoral thesis in 1965; the results were reprinted in [Forney 1966]. Forney presents serial concatenated codes of the form shown in Figure 9.17. A serial concatenated coding system uses two different codes, an outer code and an inner code. The two codes are encoded and decoded independently. Forney showed that their use can lead to an exponential decrease in error rate at the cost of more complexity and redundancy.

In current ADSL and ADSL2 standards, the inner code is a 16-state, Wei trellis code and the outer code is a Reed–Solomon code with interleaving. There is more discussion on trellis codes in Chapter 8 and on concatenated codes used for turbo coding in Chapter 10. Forney’s work focused on inner and outer block codes. Michelson and Levesque [Michelson 1985] credit Odenwalder [Odenwalder 1972] with suggesting an inner convolutional code of the type used in ADSL.

ADSL uses concatenated codes for a number of reasons. Concatenated codes can offer higher coding gains than the trellis code or the Reed–Solomon code alone. Although it would be possible to achieve the higher level of coding gain with trellis coding alone, Reed–Solomon coding is also used to combat impulse noise, as discussed in Section 9.4. When impulse noise protection is not needed or low system delay is an important consideration, the trellis code can still be used and will achieve good performance on its own.

Evaluating the performance of concatenated codes is more challenging than evaluating the performance of any single code alone. Recall how the coding gain for the Reed–Solomon code was evaluated in Section 9.3. The development started with an expression for the error rate of a q -ary (octet in the case of GF(256)) symbol (Equation 9.24) and applied the binomial theorem (Equations 9.25 and 9.26). It was assumed that q -ary errors were independent. With concatenated codes, the q -ary symbol error rate after the inner (trellis) code depends on the performance of this inner code. Furthermore, errors following the inner code often occur in bursts. Making matters still more complicated, the length of the bursts that follow the inner

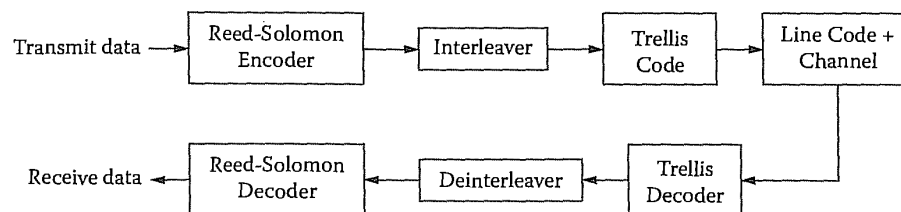


FIGURE 9.17

Serial concatenated coding used in ADSL and ADSL2. The outer code is a Reed–Solomon code, and the inner code is a 16-state trellis code.

Contents

11 Code Concatenation and Advanced Codes	258
11.1 Code Concatenation	260
11.1.1 Serial Concatenation	260
11.1.2 Parallel Concatenation	260
11.1.3 Multilevel Coding	261
11.2 Interleaving	263
11.2.1 Block Interleaving	267
11.2.2 Convolutional Interleaving	268
11.2.3 Enlarging the interpretation of triangular interleavers	272
11.2.4 Random Binary Interleaving	277
11.3 Turbo Codes	280
11.3.1 Turbo-Code rate definition	280
11.3.2 Puncturing	281
11.3.3 Analysis of probability of bit error for parallel concatenation	282
11.3.4 Analysis of probability of bit error for serial concatenation	287
11.3.5 Coding Tables for Parallel Concatenation with rate $1/n$, or for rate $n - 1/n$ with no puncturing	289
11.3.6 Parallel and Serial Turbo Code Tables with puncturing for base rate $1/2$	290
11.4 Turbo Codes for Higher-Level Constellations	297
11.4.1 The 8-state SSC Code	297
11.4.2 The 16-state PSD Code	300
11.4.3 The Third Generation Wireless Turbo Code	300
11.5 Low-Density Parity Check Codes	305
11.5.1 IBM Array LDPC Codes	306
11.5.2 Feeding the Decoder	311
11.5.3 Lee's LDPC Routines	311
Exercises - Chapter 11	317

Chapter 11

Code Concatenation and Advanced Codes

A transmission system's use of the convolutional (or block) and trellis codes of Chapter 10 allows significant improvement in performance with respect to uncoded transmission. The improvement occurs because careful selection of sequences of transmitted symbols can increase codeword separation or minimum distance for a given energy (or volume). Additional improvement through cascade of hard-coding outside of soft-coding further reduces probability of error (perhaps at a small reduction in data rate determined by the rate of the hard external code). This simple improvement is a special case of what is known as **Code Concatenation** and was first seriously studied by Forney in his 1963 MIT dissertation. Often $p = 10^{-6}$ for a BSC model, and very high-rate $\bar{b} \approx 1$ block codes (for instance Reed-Solomon codes) then can reduce probability of error to essentially zero and assume the position of the outer-most hard-decision codes.

There are many successful code-concatenation methods that are more sophisticated than simple cascade of hard and soft coding systems. While coding gains may not add directly, improved probability of bit error can occur from the concatenation, allowing essentially reliable transmission of data rates very close to capacity. Section 11.1 introduces more formally serial and parallel concatenation of codes, while Section 11.2 discusses the related concept of **interleaving**. Interleaving attempts to redistribute a burst of errors or noise that may overwhelm one of the codes, but not all the codes in a concatenation. Section 11.2 describes three popular classes of interleaving methods: **block, convolutional, and random interleaving**. The combination of two or more codes can be considered as one giant code of very long block length with an interleaver, or essentially as “randomly generated” codewords. Surprisingly, the combined code often does **not** have significantly larger free or minimum distance – however, the average number of bit errors that correspond to error events with small distance is very low. This lower number of bit errors is essentially reduced in proportion to the interleaver depth. Thus a maximum-likelihood decoder for the giant combined code could have very low probability of bit error even when the distance-to-noise ratio is poor, thus allowing reliable transmission (at some low but nonzero \bar{P}_b) at rates near capacity. However, such a maximum-likelihood decoder would be hopelessly complex.

Iterative Decoding of Section 9.6 is instead a decidedly less complex decoding strategy for concatenated codes. Rather than make a “hard” decision on any of the individual codes, “soft” information about the likelihood of different sequences and/or bits is instead computed for one code and then passed to a decoder for another code, which then also computes its own soft information. The soft information is iteratively passed between all decoders in a recursive cascade of decoders with the hope of converging close to an ability to correctly detect the transmitted message, often lowering \bar{P}_b for data rates very close to but less than capacity. As the AEP in Section 8.3 notes implicitly, any sufficiently long-block-length code with codewords chosen at random has high probability of being a capacity-achieving code so there are many interleaved code concatenations that can approach capacity levels.

Turbo Codes are addressed in Section 11.3 and are examples of interleaved concatenated codes that presume use of iterative decoding and can approach rates of capacity with low \bar{P}_b . Section 11.4 directly addresses the use of binary turbo codes in multi-level constellations. Section 11.5 investigates LDPC

codes that essentially can achieve capacity as block length is increased. In LDPC codes, the generator matrix of a block code $G(D) = G(0) = G$, or correspondingly the parity matrix $H(D) = H(0) = H$ have entries essentially chosen at random.

Section ?? discusses shaping methods. At $\bar{b} \geq .5$, the individual symbols in a sequence need to increasingly appear as if the code were created by drawing codewords from a Gaussian distribution. This effect is separate from coding gain achieved by sequence separation or nearest-neighbor reduction and is called **shaping gain** in Chapter 1. A final concatenation of shaping codes is necessary when $\bar{c} \geq .5$ to gain up to an additional 1.53 dB of coding gain that cannot be achieved any other way. Shaping codes are unusual in that most of the complexity is in the transmitter, while the receiver is trivial, and amount to selecting sequences from a large set of extended codewords so that the shaping gain is maximized.

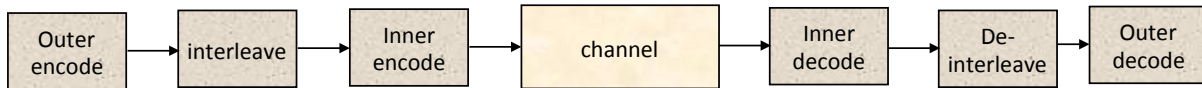


Figure 11.1: Serial Concatenation

11.1 Code Concatenation

Concatenated coding attempts to combine the performance-improvement of two or more codes. Concatenation of codes can follow one of 3 architectures, serial, parallel, or multilevel, each of which are introduced in the 3 following subsections.

11.1.1 Serial Concatenation

Figure 11.1 illustrates **serial concatenation**, which uses two codes called the **inner** and **outer** codes. The inner code may be any of the types of codes studied in earlier chapters. The outer code attempts to provide additional improvement. The interleaver rearranges the order of transmitted symbols, while the corresponding de-interleaver restores the original order. De-interleaving thus can disperse a burst of errors associated with an inner-decoder error event so that these individual errors may be more easily corrected by the outer code's decoder. De-interleaving thus allows the inner channel-code output to appear independent from symbol to symbol.

Serial concatenation with interleaving traditionally uses an inner code that makes hard decisions on its inner-code decoder output. Such hard decisions allow the inner code-and-channel to be modeled by a BSC (or DMC more generally). The outer code then is designed to reduce P_e to negligible levels: The probability of error for the binary outer code that acts on the BSC created by the inner code has

$$\bar{P}_e \approx N_e(4p)^{\lceil d_{free}/2 \rceil} . \quad (11.1)$$

Block or convolutional error-correction outer codes with high rates $\bar{b} \approx 1$ can have $d_{free} \geq 8$. Thus, the inner code's probability of error of perhaps 10^{-6} can be reduced through the use of the outer code to below 10^{-20} . Such low error probabilities essentially mean that the transmission layer is effectively error free, a highly desirable result in many applications. This simple form of serial concatenation allows one of the basic results of Shannon to be achieved, namely arbitrarily reliable transmission (albeit at some small loss in data rate not required in basic capacity theorems of Chapter 8).

Serial concatenation can also use two codes that both are designed for BSC's, meaning that the channel itself is a BSC, as well as the model of the inner-coded-and-interleaved channel. Soft decoding can be used by both decoders exploiting the equality and parity constraint viewpoints of Sections 9.5 and 9.6. Serial concatenation can be extended to cascade more than 2 codes by recursively considering the system in Figure 11.1 to be representative of a new inner code, to which a new outer code and interleaver can be applied.

11.1.2 Parallel Concatenation

Figure 11.2 illustrates parallel concatenation for two codes. The same information sequence is encoded by two different encoders. However, one of the encoders acts on an interleaved version of the input sequence. The two encoder outputs are multiplexed (and interleaved for one of the codes) for channel transport. The naming of an "inner code" and an "outer code" is somewhat vague for parallel concatenation, so the codes are often instead referred to as first and second codes. All the individual-code outputs may be transmitted, or some subset of these outputs may instead be transmitted by regular deletion or "puncturing" of the encoder outputs.

A simple form of parallel concatenation is known as a **product code**. Product codes re-encode the same bits/symbols by adding parity bits with two systematic encoders to the same set of input bits. Figure 11.3 illustrates product coding. One encoder determines horizontal parity bits while a second

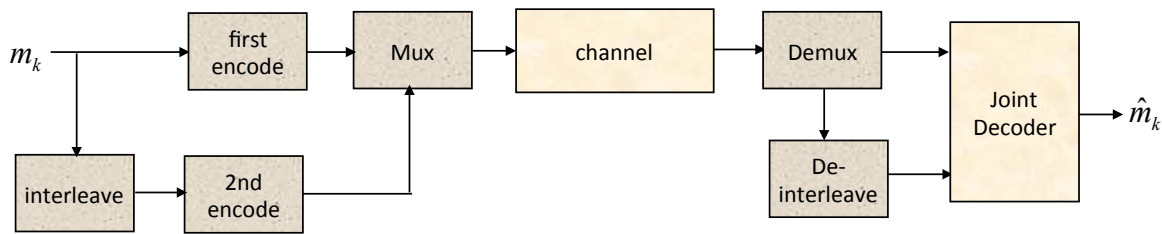


Figure 11.2: Parallel Concatenation.

encoder generates vertical parity bits.¹ The ordering of the bits for the two encoders is not the same, thus tacitly illustrating the interleaving. An incorrectly decoded codeword for the horizontal code would leave up to a single bit error in each column, which could typically still be corrected by the vertical code.

In general for parallel concatenation again, de-interleaving again distributes bursts of errors caused by the first decoder's selection of an incorrect codeword, so that the second code can more reliably decode (as long as error bursts don't occur too often). Parallel concatenation can be extended to more than two codes by adding additional interleavers and encoders in the obvious parallel way in Figure 11.2. One could envision a 3-dimensional version of the product code as an example with additional parity bits "coming out of the paper at the reader" in Figure 11.3 (20 of 27 positions in a $3 \times 3 \times 3$ cube).

11.1.3 Multilevel Coding

Multilevel coding uses partition chains to encode different bits of the input stream with different codes. Trivial forms of multilevel coding are the 4-dimensional Wei trellis codes of Chapter 10. Those codes had a first (often rate $2/3$) encoder that selected a sequence of 4-dimensional cosets to be transmitted. These first two input bits were protected by the convolutional or trellis part of the code. The remaining input bits were then used to select parallel transitions in a D_4 lattice code for each branch of the trellis. One of those latter bits is protected by the code that corresponds to selecting the upper or lower path in the trivial trellis of a D_4 lattice. The remaining bits could also be coded.

Multilevel coding will not be further pursued in this Chapter. Multilevel coding is typically applicable only to systems with very large b .

¹The 9^{th} position at the lower right in Figure 11.3 would be present in the serial concatenation of Subsection 11.1.1.

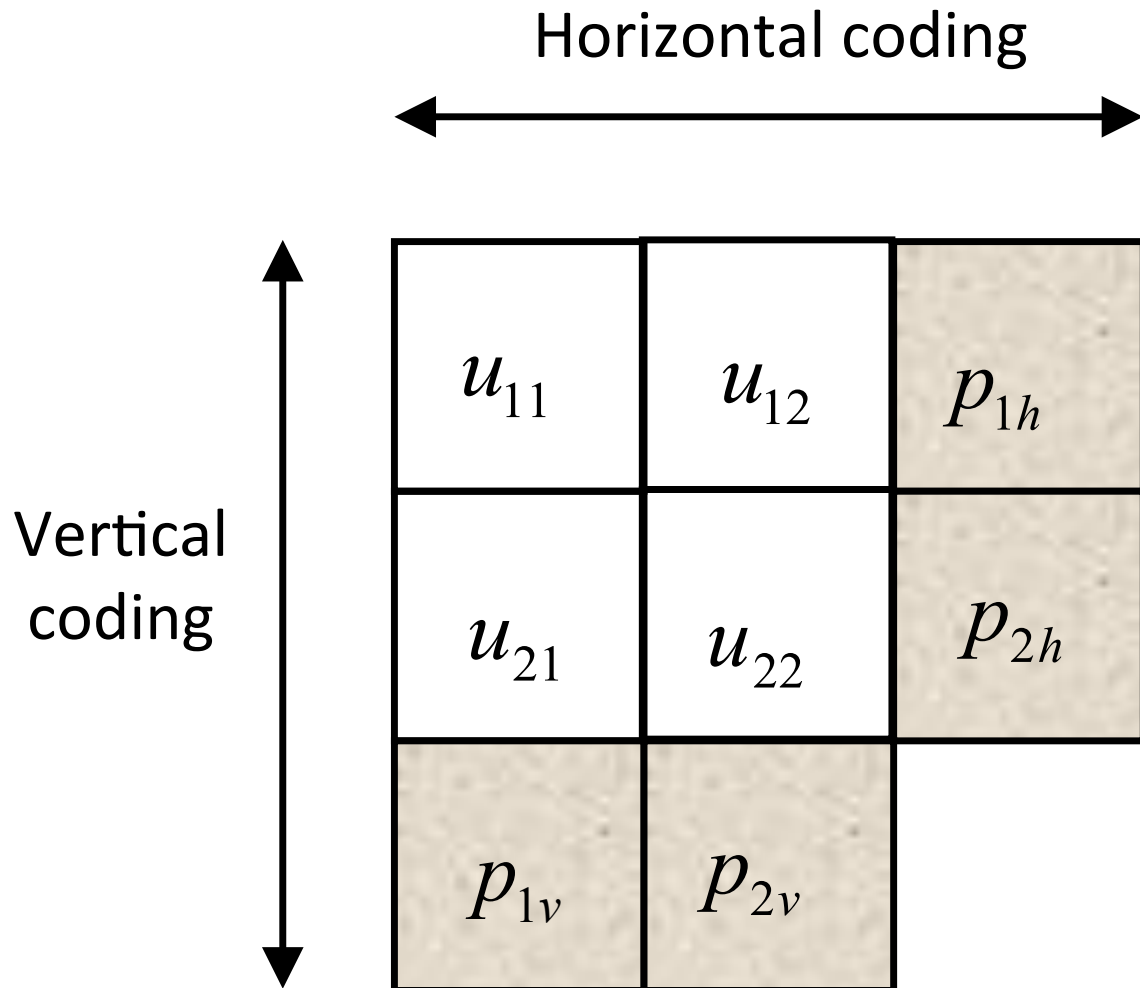


Figure 11.3: Simple product code illustration.

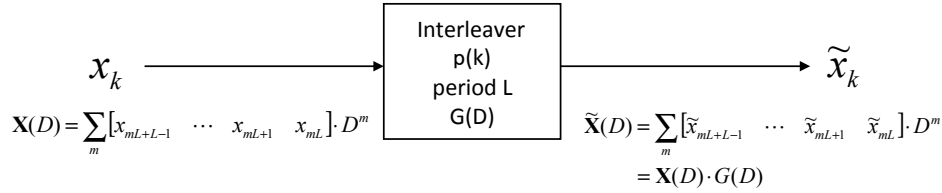


Figure 11.4: Basic Interleaver.

11.2 Interleaving

Interleaving is a periodic and reversible reordering of blocks of L transmitted symbols. Symbols (or bits) are correspondingly reordered by de-interleaving in the receiver. Interleaving is used to disperse error bursts that may occur because of nonstationary channel noise that may be localized to a few dimensions such as an impulse noise burst in time or a loss of a narrow frequency band in OFDM. Error bursts can also occur because of the incorrect decision of an inner/first decoder in Figures 11.1 or 11.2. Errored symbols caused by inner-code detection errors will typically span the entire incorrect codeword, leading to a burst of errored symbols. If these bursts are separated by an interval long with respect to the interleaver period, then they can be distributed more evenly over time (or more generally dimensions) by the de-interleaver in the receiver. The distribution of the errors effectively enables realistic modeling for the inner-code-and-channel as memoryless, i.e., modeled by a BSC, DMC, or other channel for which successive symbol outputs are independent.

Figure 11.4 generically depicts the interleaver as accepting symbols indexed in time by k or in block/packet in m , where L symbols occur within one packet and L is the period of the interleaver. Often L corresponds to a codeword size.

Definition 11.2.1 (Depth of an Interleaver) *The depth J of an interleaver is defined as the minimum separation in symbol periods at the output of the interleaver between any two symbols that were adjacent at the input of the interleaver.*

The depth of an interleaver has significant implication for a burst of errors entering a de-interleaver at a receiver. If a burst of errors has duration less than the depth, then two symbols affected by the burst cannot be adjacent after de-interleaving.

Definition 11.2.2 (Period of an Interleaver) *The period L of an interleaver is the shortest time interval for which the re-ordering algorithm used by the interleaver repeats.*

Essentially, the period is established by the detailed description of the interleaver and measures the length of a block of input symbols to which interleaving is applied. The interleaver repeatedly acts with the same algorithm upon successive blocks of L symbols. Often the period of the interleaver is chosen to be equal to the block length of an outer code when block codes are used in serial concatenation.

Theorem 11.2.1 (Minimum Distance Magnification in Interleaving (for serial concatenation))

For inner channels with an outer hard-decision code of block length equal to the period of interleaving, and only one error burst occurs within $(J - 1) \cdot (L - 1)$ symbols, then the outer code's free distance in symbols is multiplied by the depth J for inner-channel error bursts.

proof: *Since all adjacent de-interleaver output symbols within a burst on the input to the de-interleaver are separated by at least $J - 1$ symbols, the number of errors has been effectively reduced by a factor of J as long as a subsequent or preceding burst does not introduce errored*

symbols into the same codewords that have errors from the burst under investigation. The minimum delay of any symbol through the process of interleaving and de-interleaving is $(J - 1) \cdot (L - 1)$ symbols (and would occur when all adjacent symbols are always spaced by exactly $J - 1$ symbols after interleaving) if every one of L symbols is spaced by $J - 1$ symbols from its previous neighbors. Thus, the burst length must be in general no longer than this minimum delay to prevent different error bursts from placing errored symbols in the same de-interleaver-output codewords.

Interleaving with a single code does not improve performance for stationary additive white Gaussian noise, so it is used for channels that exhibit bursts of errors or nonstationary noise. A burst of errors can occur when an “inner” decoder incorrectly decodes and thus interleaving can be of value in systems with two codes as in section 11.3.

Generally, the interleaver follows a relationship from its input \mathbf{x}_k to its output $\tilde{\mathbf{x}}_k$ of

$$\tilde{\mathbf{x}}_k = \mathbf{x}_{\pi(k)} \quad , \quad (11.2)$$

where $\pi(k)$ is a function that describes the mapping of interleaver output time indices to interleaver input time indices. Necessarily $\pi(k)$ is one-to-one over the integers modulo its period of L samples. Because of the periodicity,

$$\pi(k) - L = \pi(k - L) \quad . \quad (11.3)$$

The depth can be more precisely defined mathematically using the function π as

$$J = \min_{k=0, \dots, L-1} | \pi^{-1}(k) - \pi^{-1}(k+1) | \quad . \quad (11.4)$$

This section augments the traditional coding use of a delay variable D as corresponding to one interleaver period by considering also a symbol delay variable D_{sym} such that

$$D = D_{sym}^L \quad . \quad (11.5)$$

As far as the author knows, no other text uses the symbol delay notation, but we find it very useful in simplifying the description of interleaving.

In traditional study of interleaving, a sequence of interleaver-input symbols can be represented by the L -dimensional (row) vector sequence $\mathbf{X}(D)$ in Figure 11.4 where each element in the sequence spans one period of the interleaver, with the time index $k = m \cdot L + i$ $i = 0, \dots, L - 1$, yielding to an interleaver block index m ,

$$\mathbf{X}_m = [\mathbf{x}_{m \cdot L + (L-1)} \ \mathbf{x}_{m \cdot L + (L-2)} \ \dots \ \mathbf{x}_{m \cdot L}] \quad , \quad (11.6)$$

where m thus denotes a time index corresponding to a specific block of L successive interleaver-input symbols, and

$$\mathbf{X}(D) = \sum_m \mathbf{X}_m D^m \quad . \quad (11.7)$$

Similarly the interleaver output $\tilde{\mathbf{X}}(D)$ can be considered an L -symbol-element sequence of interleaver outputs. Then, interleaving can be modeled as a “rate 1” convolutional/block code over the symbol alphabet with generator and input/output relation

$$\tilde{\mathbf{X}}(D) = \mathbf{X}(D) \cdot G(D) \quad , \quad (11.8)$$

where $G(D)$ is an $L \times L$ nonsingular generator matrix with the following restrictions:

1. one and only 1 entry in each row/column can be nonzero, and
2. nonzero entries are of the form D^l where l is an integer.

The de-interleaver has generator $G^{-1}(D)$, so that $\mathbf{X}(D) = \tilde{\mathbf{X}}(D) \cdot G^{-1}(D)$. Further, a **causal interleaver** has the property that all nonzero elements have D^l with $l \geq 0$, and elements above the diagonal must have $l \geq 1$ – the de-interleaver for a causal interleaver is necessarily noncausal, and thus must be

instead realized with delay because interleaving necessarily introduces delay from interleaver input to de-interleaver output.

The equivalent relationships in terms of symbol delay are expressed similarly

$$\tilde{\mathbf{X}}(D_{sym}) = \mathbf{X}(D_{sym}) \cdot G(D_{sym}) \quad , \quad (11.9)$$

but the input and output vectors have each entry defined in terms of the symbol-period D -transform as

$$x_i(D_{sym}) = \sum_{k=0}^{\infty} x_{i,k} \cdot D_{sym}^k \quad . \quad (11.10)$$

The entries in the vector $\mathbf{X}(D_{sym})$ are **not** simply found by substituting $D = D_{sym}^L$ into the entries in the vector $\mathbf{X}(D)$. In fact,

$$\mathbf{X}(D_{sym}) = \left[D_{sym}^{L-1} \cdot x_{L-1}(D) \Big|_{D=D_{sym}^L} \quad D_{sym}^{L-2} \cdot x_{L-2}(D) \Big|_{D=D_{sym}^L} \quad \dots \quad x_0(D) \Big|_{D=D_{sym}^L} \right] \quad . \quad (11.11)$$

A similar relation to (11.11) holds for the output vector of the interleaver. The symbol-spaced generator follows easily by replacing any nonzero power of D , say D^δ in $G(D)$ by substituting $D_{sym}^{L \cdot \delta + (\text{column number} - \text{row number})_{mo}}$ and circularly shifting the entry within the row to the left by (row number – column number) positions.

Rule 11.2.1 (Generator Conversion for Interleavers) Let any non-zero entry in $G(D_{sym})$ be written as D_{sym}^r . A new column index i' is formed by $i' = (r + i)_{\text{mod } L}$ where i is the column index of the original non-zero entry (counting from right to left, starting with 0). Further let $r' = \lfloor \frac{r+i}{L} \rfloor$, then place $D^{r'}$ in the same row in the position of column i' . One can think of this as circularly shifting by $r + i$ positions and increasing the power of D for every factor of L in the exponent of D_{sym}^{r+i} .

(An example use of Rule 11.2.1 will occur shortly.)

It is often easier to avoid the rule and simply directly write $G(D)$ based on a description of the interleaving rule (and to do the same for $G(D_{sym})$). A few examples will help illustrate the concepts.

EXAMPLE 11.2.1 (Simple Block Permutation) A simple period-3 HW example is

$$\pi(k) = \begin{cases} k+1 & \text{if } k = 0 \bmod 3 \\ k-1 & \text{if } k = 1 \bmod 3 \\ k & \text{if } k = 2 \bmod 3 \end{cases} \quad (11.12)$$

or in tabular form:

$\pi(k):$	-1	1	0	2	4	3	5
$k:$	-1	0	1	2	3	4	5

which has inverse de-interleaver easily expressed as

$k' = \pi(k):$	-1	0	1	2	3	4	5
$\pi^{-1}(k') = k:$	-1	1	0	2	4	3	5

The depth is $J = 1$ and thus clearly this interleaver is not very useful.

The corresponding generator is:

$$G(D) = G(D_{sym}) = G(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (11.13)$$

with de-interleaving inverse

$$G^{-1}(D) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad . \quad (11.14)$$

Note that the two interleaver descriptions are equivalent if there are only binary entries in the generator matrix. Such interleavers are called “block interleavers” and studied in Section 11.2.1.

EXAMPLE 11.2.2 (Simple Triangular Interleaving) A second period-3 HW-interleaver example is

$$\pi(k) = \begin{cases} k & \text{if } k = 0 \bmod 3 \\ k - 3 & \text{if } k = 1 \bmod 3 \\ k - 6 & \text{if } k = 2 \bmod 3 \end{cases} \quad (11.15)$$

or in tabular form:

$\pi(k):$	-7	0	-2	-4	3	1	-1
$k:$	-1	0	1	2	3	4	5

with inverse

$\pi(k):$	-1	0	1	2	3	4	5
$k:$	5	0	4	8	3	7	11

The depth of this interleaver is $J = 4$ symbol periods.

The generator for the second HW example is (recall a delay, D , corresponds to a delay of one period of $L = 3$ time samples)

$$G(D) = \begin{bmatrix} D^2 & 0 & 0 \\ 0 & D^1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad G(D_{sym}) = \begin{bmatrix} D_{sym}^6 & 0 & 0 \\ 0 & D_{sym}^3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11.16)$$

This interleaver is clearly a one-to-one (nonsingular) generator and has inverse

$$G^{-1}(D) = \begin{bmatrix} D^{-2} & 0 & 0 \\ 0 & D^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (11.17)$$

This block-level inverse requires a delay of 2 block periods or 6 symbol periods for causal inversion.

EXAMPLE 11.2.3 (A depth-3 Interleaver) An interleaver with period $L = 5$ and depth $J = 3$ has generator

$$G(D_{sym}) = \begin{bmatrix} D_{sym}^8 & 0 & 0 & 0 & 0 \\ 0 & D_{sym}^6 & 0 & 0 & 0 \\ 0 & 0 & D_{sym}^4 & 0 & 0 \\ 0 & 0 & 0 & D_{sym}^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.18)$$

and could be described as delaying each symbol within a period by its index times $(J - 1)$ symbol periods. The inverse is also easily described. Using the traditional notation with D corresponding to a block period, then rule 11.2.1 provides

$$G(D) = \begin{bmatrix} 0 & 0 & D^2 & 0 & 0 \\ D & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11.19)$$

The second description often complicates easy insight into the implementation and the inverse. The interpretation that each symbol is delayed by its index times $J - 1$ symbol periods is not nearly as evident (although true if one examines much harder). This simplified insight is the value of using the symbol-spaced interpretation, and the nice $D_{sym}^{(J-1)i}$ diagonal in (11.18)

11.2.1 Block Interleaving

Block interleaving or **permutation interleaving** is the simplest type of interleaving and has $G(D) = G(D_{sym}) = G(0) = G$. The de-interleaver is trivially $G^{-1} = G^*$ for the block interleaver. The permutation of inputs to outputs is contained within one period in a block interleaver. The first HW example in (11.13) is a block interleaver. Block interleavers are never causal unless $G = I$ (and $G = I$ trivially means no interleaving).

Figure 11.5 illustrates one of the most often encountered block interleavers, which is most often associated with serial code concatenation and a block inner code. Each successive inner-code block of symbols (often a codeword for a block code, but not necessarily the case) is written into a corresponding register/row in the interleave memory. the block interleaver outputs instead successive columns. The number of symbol blocks stored is the depth J of the interleaver. If a symbol group has K symbols and the interleaver depth is J , then $K \cdot J$ symbols must be stored in each of two transmit memories of a block interleaver as shown for $J = 3$ and $K = 4$. The period is $L = K \cdot J$. As $K = 4$ symbols at a time are written into each row of one of the transmit memory buffers, $J = 3$ symbols in each column are read from the other. Symbols occur every T seconds, making the symbol rate $1/T$. The interleaver input clock is thus $1/K = 1/4$ of the symbol clock rate. Thus, one symbol block of $K = 4$ symbols is written into a each row of the write-memory buffer. After $L = K \cdot J = 12$ symbol periods, the entire write buffer will be full. The transmit-memory output clock is $1/J = 1/3$ of that same symbol clock. The read-memory buffer is read $J = 3$ symbols from each column for each period of the interleaver output clock. The interleaver starts/completes writing of the write buffer at exactly the same two points in time as it starts/completes reading the read buffer. Every $L = KJ = 12$ symbol-clock periods, the read and write memories are interchanged.

Ignoring, the 12-symbol-period delay necessary for causal implementation, the generator for this block interleaver (assuming entire rows/columns are read in sequence, i.e., there are no shift-registers)

$$G(D) = G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11.20)$$

The de-interleaver is G' (where prime means transpose) in the receiver and accepts symbols from a decoder and writes them in terms of K symbols per column successively. After all $L = KJ$ symbols have been stored, the symbol blocks are read in horizontal or row order. Again, two memories are used with one being written while the other is read. This type of interleaver is sometimes called a classical block interleaver. For this classical form, the end-to-end delay is (no worse than) $2L$ symbol times and correspondingly there is a total of $4L$ RAM locations necessary (at receiver and transmitter), half ($2L$) of which are in the receiver and the other half ($2L$) in the transmitter.

For the classical block interleaver, a burst of B errored symbols is distributed roughly evenly over J symbol blocks by the de-interleaving process in the receiver. If this is the only burst within the total L receiver symbols and the symbol block length is equal to the length of a codeword for a block code, then the outer code with hard decisions can correct approximately J times more errored symbols. Larger depth J means more memory and more delay, but greater power of the outer code as long as a second burst does not occur within the same KJ symbols. The minimum distance of the code is thus essentially multiplied by J as long as errors are initially in bursts that do not occur very often. Thus interleaving easily improves the performance of concatenation.

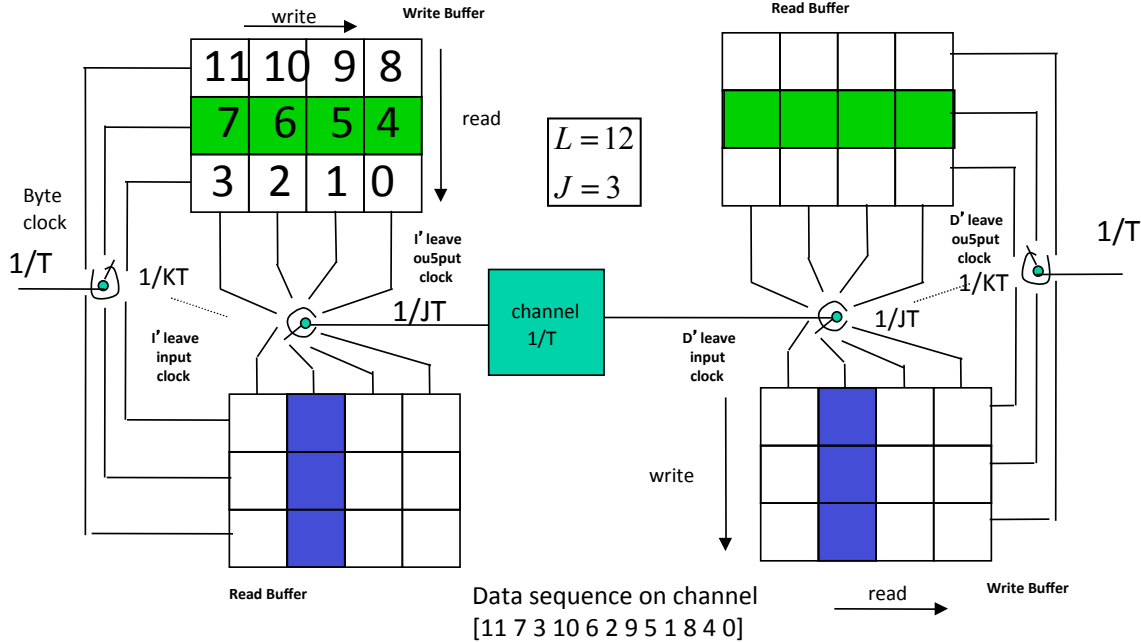


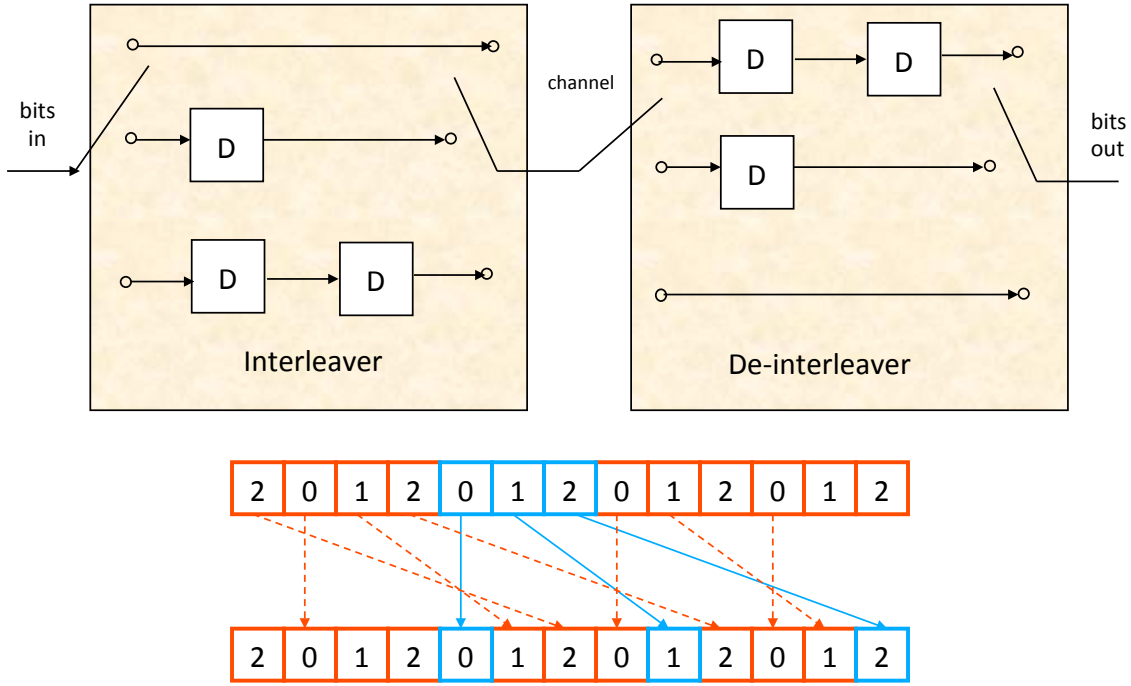
Figure 11.5: Illustration of classical block interleaving and de-interleaving with $L = 12$ and $J = 3$.

11.2.2 Convolutional Interleaving

Convolutional interleavers have $G(D) \neq G(0)$, meaning there is at least one delay element. While more complex in concept, convolutional interleavers can allow a reduction in delay and in memory required for implementation of a given depth J .

Coding theorists have often reserved the name **triangular interleaver** for the special case of a convolutional interleaver where $G(D)$ is a diagonal matrix of increasing or decreasing powers in D proceeding down the diagonal. Example 11.2.2 illustrated such a triangular interleaver. The reason for the names “multiplexed” or “triangular” interleaver follow from Figure 11.6, which illustrates the 3×3 implementation. The delay elements (or the memory) are organized in a triangular structure in both the interleaver and the de-interleaver. Symbols enter the triangular interleaver from the left and with successive symbols cyclically allocated to each of the 3 possible paths through the interleaver in periodic succession. The input switch and output switch for the interleaver are synchronized so that when in the upper position, the symbol simply passes immediately to the output, but when in the second (middle) position the symbol is stored for release to the output the next time that the switch is in this same position. Finally, the bottom row symbols undergo two interleaver periods of delay before reaching the interleaver output switch. The deinterleaver operates in analogous fashion, except that the symbol that was not delayed at the transmitter is now delayed by two periods in the receiver, while the middle symbol that was delayed one period in transmitter sees one additional period of delay in receiver, and the symbol that was delayed twice in transmitter is not delayed at all at receiver. Clearly all symbols then undergo two interleaver periods of delay, somehow split between transmitter and receiver. Any symbols in the same block of 3 on the input have at least 3 symbols from other blocks of inputs in between as illustrated in Figure 11.6. The depth is thus $J = 4$.

To generalize what has been known as the triangular interleaver, the depth is restricted to be equal to $J = L + 1$, and $K = L$. Then, $K = L$ symbols within the m^{th} symbol block are numbered from $x_{0,m}$, ..., $x_{i,m}$, ..., $x_{L-1,m}$ and the triangular interleaver delays each such symbol by $i \cdot L = i(J - 1)$ symbol

Figure 11.6: Triangular interleaver illustrated for $L = 3$.

periods. The generator is

$$G(D) = \begin{bmatrix} D^{L-1} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & D & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix}. \quad (11.21)$$

The diagonal increasing power of D (where again D corresponds to L symbol periods of delay) in $G(D)$ is that the i^{th} symbol in a symbol block is delayed

$$\Delta_i = i \cdot L = i \cdot (J - 1) \text{ symbol periods } i = 0, \dots, L - 1 \quad (11.22)$$

symbol clocks, assuming symbols arrive sequentially. The single block-of-symbols delay D in Figure 11.6 would be 3 symbol delays, so $D = D_{sym}^3$ and is realized as one storage element. The depth is clearly $J = L + 1$ because the increment in delay between successive symbols is D_{sym}^{J-1} .

The straightforward deinterleaver inverse $G^{-1}(D)$ has negative powers of D in it, so must be realized with delay of $L(L - 1) = (J - 1)(L - 1) = L^2 - L$ symbol periods to be causal. This is equivalent to multiplying $G^{-1}(D)$ by D^{L-1} to obtain

$$G_{causal}^{-1}(D) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & D^{L-2} & 0 \\ 0 & 0 & \dots & D^{L-1} \end{bmatrix}. \quad (11.23)$$

The total delay is $L(L - 1) = (J - 1)(L - 1) = L^2 - L$ symbol periods, which is the theoretical minimum possible. Clearly, the triangular interleaver requires only at most 1/4 the memory and exactly 1/2 the delay of the classical block interleaver, but carries the restriction (so far) of $J = L + 1$ and $K = L$.

In triangular interleaving, the period L can be set equal to the codeword length. The period is no longer the product of K and J as in the block interleaver. Because of the shorter period, synchronization

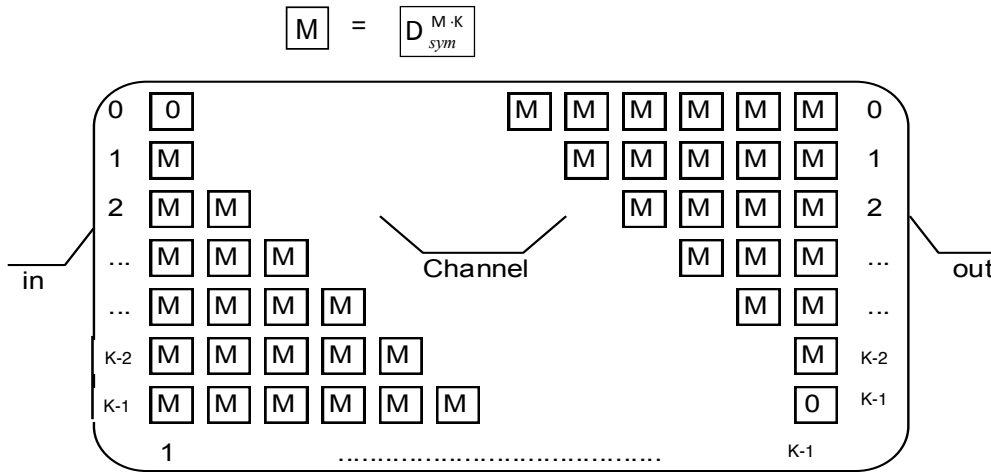


Figure 11.7: The Generalized Triangular Interleaver.

to boundaries of $K \cdot J$ blocks of symbols is no longer necessary (although clearly the de-interleaver in the receiver still needs to know the $L = J - 1$ symbol boundaries of the interleaver). For this reason, the triangular interleaver is often said to not require synchronization – this is somewhat of a misnomer, in that it requires less synchronization than the block interleaver.

The triangular interleaver has an overly restrictive depth of $J = L + 1$ only, but has an attractive and simple triangular implementation. **Generalized triangular** interleaving releases this depth constraint somewhat by essentially replacing each delay element in the triangular implementation by a FIFO (first-in-first-out) queue of M symbol periods. The period remains L symbol periods, but the depth increases to $J = M \cdot L + 1$ symbol periods.

The generalized triangular interleaver appears in Figure 11.7 where the box with an M in it refers to a FIFO containing M symbols and a delay of $M \cdot L$ symbol periods (or M interleaver periods of L symbol periods each). The generalized triangular interleaver is designed for use with block codes of length $N = q \cdot L$ where q is a positive integer. Thus a codeword is divided into q interleaver-period-size groups of size L symbols that are processed by the generalized triangular interleaver. If $q = 1$, one codeword is processed per period, but otherwise $(1/q)^{th}$ of a codeword in general. Often, a symbol is a byte (or “octet” of 8 bits as stated in International Telecommunications Union (ITU) standards).

Thus, the delays for this particular interleaver can be written as $D_{sym}^{M \cdot L}$. The i^{th} symbol entering the interleaver in Figure 11.7 is delayed by $i \cdot L \cdot M$ block periods. The total delay of any byte is then $\Delta = L \cdot (L - 1) \cdot M = (J - 1) \cdot (L - 1)M$ symbol periods. As with all triangular interleavers, the delay element has power $J - 1$ so $J = M \cdot K + 1$ is again the depth.

The generalized triangular interleaver is typically used with a block code that can correct t symbols in error (typically Reed Solomon block code where a symbol is a byte, then t is the number of parity bytes if erasures are used and $1/2$ the number of parity bytes if no erasures are used). The block-code codewords may thus be subdivided into q blocks of symbols so that $N = q \cdot K$. Then Table 11.1 lists the various parameter relations for the generalized triangular interleaver.

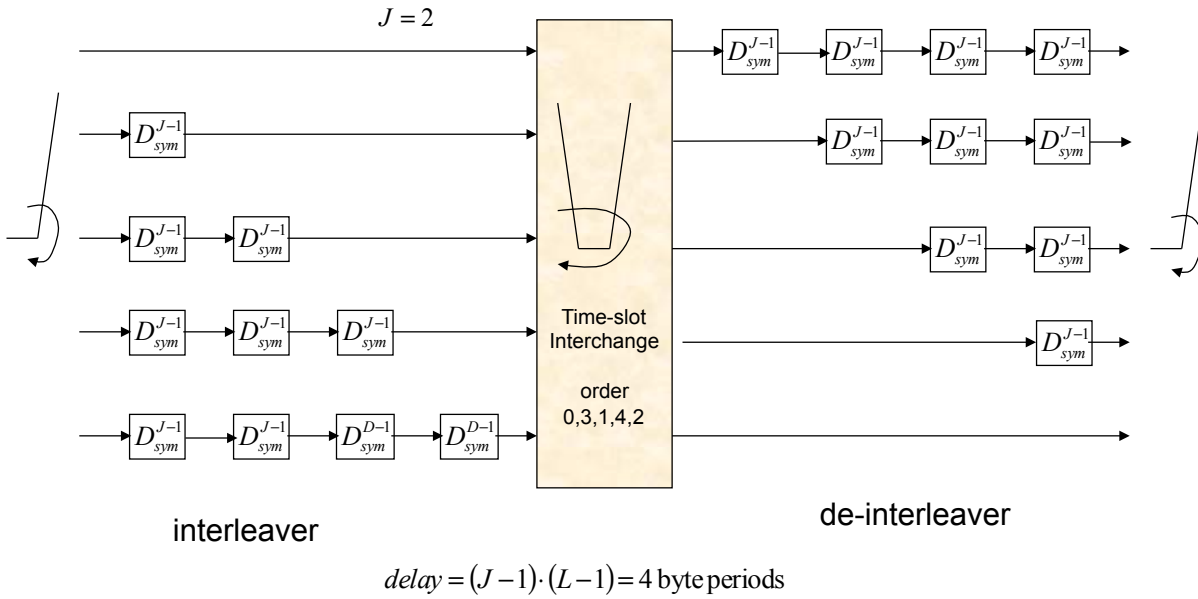
EXAMPLE 11.2.4 (VDSL Generalized Triangular Interleaver) The ITU G.993.1 (G.993.2) VDSL (VDSL2) standards use DMT with a Generalized Triangular Interleaver. Table 11.2 works some specific data rates, interleave depths, and consequent delays and memory sizes for the VDSL generalized triangular interleaver. More delay can create application problems, but also improves the correcting capability of the code by the depth parameter.

Parameter	Value
Interleaver block length (K)	$K=L$ symbols (equal to or divisor of N)
Interleaving Depth (J)	$J=M \lceil \frac{K}{L} \rceil$
(De)interleaver memory size	$M \lceil \frac{K}{L} \rceil \lceil \frac{K-1}{2} \rceil$ symbols
Correction capability (block code that corrects t symbol errors) With $q=N/K$	$\lceil \frac{t}{q} \rceil \lceil \frac{K}{L} \rceil \lceil \frac{M \lceil \frac{K}{L} \rceil K+1}{2} \rceil$ symbols $\lceil \frac{t}{q} \rceil \lceil \frac{K}{L} \rceil \lceil \frac{J}{2} \rceil$
End to end delay	$M \lceil \frac{K}{L} \rceil \lceil \frac{K-1}{2} \rceil$ symbols

Table 11.1: Table of parameters for Generalized Triangular Interleaver.

Rate (kbps)	Interleaver parameters	Interleaver depth (J)	(De)interleaver memory size	Erasure correction	End-to-end delay
50x1024	$K = 72$ $M = 13$	937 blocks of 72 bytes	33228 bytes	3748 bytes 520 ms	9.23 ms
24x1024	$K = 36$ $M = 24$	865 blocks of 36 bytes	15120 bytes	1730 bytes 500 ms	8.75 ms
12x1024	$K = 36$ $M = 12$	433 blocks of 36 bytes	7560 bytes	866 bytes 501 ms	8.75 ms
6x1024	$K = 18$ $M = 24$	433 blocks of 18 bytes	3672 bytes	433 bytes 501 ms	8.5 ms
4x1024	$K = 18$ $M = 16$	289 blocks of 18 bytes	2448 bytes	289 bytes 501 ms	8.5 ms
2x1024	$K = 18$ $M = 8$	145 blocks of 18 bytes	1224 bytes	145 bytes 503 ms	8.5 ms

Table 11.2: Some calculated delays and data rates for a symbol equal to a byte in the VDSL Interleaver.

Figure 11.8: Triangular interleaver illustrated for $K = 5$ and $J = 2$.

11.2.3 Enlarging the interpretation of triangular interleavers

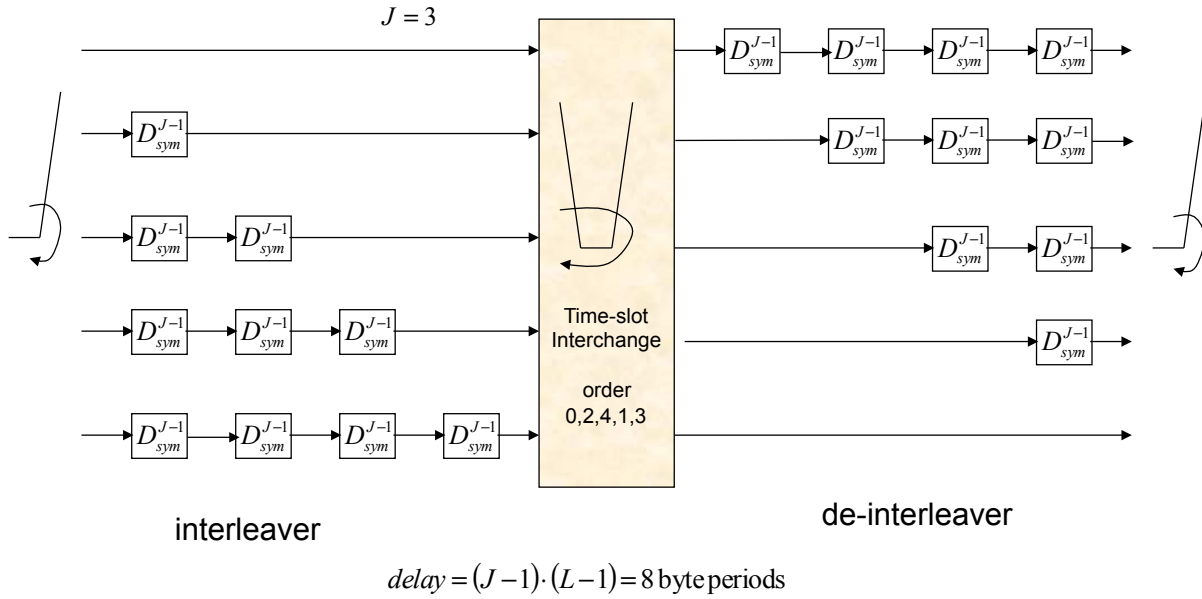
Equation (11.22) suggests a more general form of triangular interleaving that follows when $J \leq L + 1$, which is common in use, as shown in Figure 11.8 for example for $J = 2$ and $L = 5$. The delay of each symbol after it enters the interleaver is again $\Delta_i = i \cdot (J - 1) \forall i = 0, \dots, L - 1$ **symbol** periods. In Figure 11.8, this delay simply increments with each symbol because $J - 1 = 1$. The triangular structure is maintained, but the interleaver-output clocking is irregular as also shown in Figure 11.8 and also in Figure 11.9 for two different depths and a period of 5 symbols. Many authors call the interleavers with $J < L + 1$ a “convolutional interleaver,” using the more general term because the triangular implementation escaped notice. The triangular implementation follows by noting the generator in form $G(D_{sym})$ remains diagonal with decreasing (increasing) powers of D_{sym} , while the generator $G(D)$ has a more complicated non-diagonal form. Thus, the first symbol ($i = 0$) in a symbol block is not delayed at all, while the second symbol is delayed by $J - 1$ **symbol** periods, and the last symbol is delayed by $(L - 1)(J - 1)$ symbol periods. The generator matrix for the $J = 3$ example in Figure 11.9 appeared in the third example earlier. It had a complicated $G(D)$, but a simple $G(D_{sym})$. For the example with $J = 2$ of Figure 11.8, the generator is

$$G(D) = \begin{bmatrix} 0 & D & 0 & 0 & 0 \\ 0 & 0 & 0 & D & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.24)$$

or

$$G(D_{sym}) = \begin{bmatrix} D_{sym}^4 & 0 & 0 & 0 & 0 \\ 0 & D_{sym}^3 & 0 & 0 & 0 \\ 0 & 0 & D_{sym}^2 & 0 & 0 \\ 0 & 0 & 0 & D_{sym} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.25)$$

The diagonal form again is simpler and corresponds directly to the implementation. The interior order is different from the case when $J = L + 1$ and is shown as a “time-slot interchange” in Figures 11.8 and 11.9. The order is easily derived: The input switch position to the interleaver (and output of the

Figure 11.9: Triangular interleaver illustrated for $K = 5$ and $J = 3$.

de-interleaver) cycles through the period in the normal manner with index $k = 0, \dots, L - 1$. The total delay of the k^{th} symbol with respect to the beginning of the period on any line i is $i + i(J - 1) = iJ$ symbol periods. After this delay of iJ symbol periods, the symbol must leave the interleaver and thus the interleaver output switch position (and also the de-interleaver input position) must be then be such at time k that

$$(iJ)_L = k \quad , \quad (11.26)$$

since k is also measured from the beginning of an interleaver period. That is, at time k , the output switch position is a function of k , $i(k)$, such that this equation is satisfied for some index i . When the equation is solved for time $k = 1$, let us call that particular time-one solution $i(1) = \Delta$, and $(\Delta J)_L = 1$. For all other times k , then the output position is

$$i = (k \cdot \Delta)_L \quad , \quad (11.27)$$

which is easily proved by substituting (11.26) into (11.27) or $((J \cdot i)_L \cdot \Delta)_L = ((J \cdot \Delta)_L \cdot i)_L = (1 \cdot i)_L = i$. The switch orders in Figure 11.8 both satisfy this equation for the particular depth. Any depth $J \leq L + 1$ is possible unless J and L have common factors. If J and L have common factors, then (11.26) does not have a unique solution for each value of i , and the interleaver is no longer a 1-to-1 transformation. The delay of this triangular interleaver and de-interleaver (with time-slot interchange) is then always $(J - 1)(L - 1)$ symbol periods. The astute reader may note, however, that the memory requirement in the straightforward implementation shown is excessive. The memory can be easily reduced to the theoretical minimum of $(J - 1)(L - 1)/2$ in each of the interleaver and de-interleaver by a memory-reuse algorithm described momentarily.

The generalized triangular interleaver will follow exactly the same procedure for any depth $J \leq M \cdot L + 1$. Since M can be any positive integer, then any interleave depth is thus possible (as long as L and J are co-prime). The time-slot algorithm still follows (11.27). Again, memory in the straightforward conceptual implementation is not minimum, but delay is again at the minimum of $(J - 1)(L - 1)$ symbol periods.

Memory reduction to the theoretical minimum

A couple of examples in Tables 11.3 and 11.4 illustrate, for the situations of Figure 11.8 and Figure 11.9, the situation in the triangular interleaver now generalized where RAM cells can be reused. For the

Table 1 for $J=2$ and $L=5$												
Line/time	0	1	2	3	4	0'	1'	2'	3'	4'	0''	1''
0	-	-	-	-	-	-	-	-	-	-	-	-
1	-	B1	-	-	-	-	B1'	-	-	-	-	B1''
2	-	-	B2	B2	-	-	-	B2'	B2'	-	-	-
3	-	-	-	B3	B3	B3	-	-	B3'	B3'	B3'	-
4	-	-	-	-	B4	B4	B4	B4	-	B4'	B4'	B4'
CELL1	-	B1	B2	B2	B4	B4	B4	B4	B3'	B3'	B3'	B3'
CELL2	-	-	-	B3	B3	B3	B1'	B2'	B2'	B4'	B4'	B4'

Table 11.3: Minimum-memory scheduling for the triangular interleaver of Figure 11.8 with $L = 5$ and $J = 2$.

situation of Figure 11.8 with $J = 2$, the theoretical minimum number of memory cells for interleaving is 2, while the triangular illustration in Figure 11.8 uses 10 memory cells. Table 11.3 illustrates the storage of symbols (which are bytes in this example). Time is indexed over 3 successive periods of interleaving, with no prime for first period, single prime for second period, and double prime for the two byte intervals shown in the third period. A byte is indexed by the period in which it occurred as B0, B1, B2, B3, or B4 with primes also used. Line 0's bytes (B0) is always immediately passed at time 0 and therefore never uses any memory, and thus does not appear in the table. Hyphens indicate "idle" memory. After byte time 3 of the first interleaver period, the interleaver is in steady state and there are never more than 2 bytes stored at any time (presuming the interleaver reads each memory location before writing any memory location on each byte time slot). Thus in fact two memory cells could be used for this triangular/convolutional interleaver. One half of the bytes in time 1 of the interleave period (called B1 with various primes) are in CELL1, while the other of those bytes are in CELL2. This is true for all bytes, and in general, $1/J$ of the bytes in any symbol position within a period are in any particular cell. Once steady state is reached, all cells are always full. The de-interleaver also only needs 2 CELLS of memory and would be described by letting $B_i = B(L-1-i)$ everywhere (so B4 passes immediately and then bytes B3, B2, B1, and B0 undergo linearly increasing delay).

Table 11.4 shows a similar situation for $J = 3$. After time 1 of the third period, the interleaver is in steady state and uses all the minimum of $4 = \frac{(J-1)(L-1)}{2}$ memory cells. Each memory cell progressively stores the symbols from line 1, then line 4, then line 3, and line 2 before rotating back to line 1 again. The process is regular and repeats on the different memory CELLS offset in time by one period with respect to one another.

An easy way to determine a schedule for the use of the minimum number of memory cells is to realize that the same cell that is read on any byte period of any period must also be written with the next available byte of input with minimum RAM. At design time for a particular specified depth and period, a set of

$$\text{minimum number of cells} = \frac{(J-1)(L-1)}{2} \quad (11.28)$$

"fake" RAM cells can be created in computer software, each with a time that is set to "alarm" exactly $k(J-1)$ symbol periods later where k is the interleaver-input byte-clock index. At each subsequent time period in steady state, one and only one cell's timer will alarm, and that cell should be read and then written and the timer reset to the value of $k(J-1)$. Schedules of "which byte when" will then occur for each storage cell that can be stored and used in later operation. This schedule will repeat over an

Table 2 for $J=3$ and $L=5$															
L/t	0	1	2	3	4	0'	1'	2'	3'	4'	0''	1''	2''	3''	4''
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	B1	B1	-	-	-	B1'	B1'	-	-	-	B1''	B1''	-	-
2	-	-	B2	B2	B2	B2	-	B2'	B2'	B2'	B2'	-	B2''	B2''	B2''
3	-	-	-	B3	B3	B3	B3	B3	B3	B3'	B3'	B3'	B3'	B3''	B3''
4	-	-	-	-	B4	B4	B4	B4	B4	B4	B4	B4	B4'	B4'	B4''
CELL1	-	B1	B1	B3	B3	B3	B3	B3	B3	B4'	B4'	B4'	B4'	B4'	B4'
CELL2	-	-	B2	B2	B2	B2	B1'	B1'	B3'	B3'	B3'	B3'	B3'	B3'	B4''
CELL3	-	-	-	-	B4	B4	B4	B4	B4	B4	B4	B4	B2''	B2''	B2''
CELL4								B2'	B2'	B2'	B2'	B1''	B1''	B3''	B3''

Table 11.4: Minimum-memory scheduling for the triangular interleaver of Figure 11.8 with $L = 5$ and $J = 3$.

interval for each cell no longer than

$$S = \text{cell schedule length} \leq \sum_{i=1}^{L-1} i \cdot (J-1) = \frac{1}{2}(J-1) \cdot L \cdot (L-1) = \frac{\Delta}{2} \cdot L \quad (11.29)$$

symbol periods for each cell. Equality occurs in (11.29) when

$$\frac{m \cdot L}{J-1} \notin Z \text{ for any } m < \frac{\Delta}{2} - 1 \quad (11.30)$$

When equality occurs then any integer number of periods less than S cannot be divided by the interleaver-output spacing between formerly adjacent interleaver input symbols $J-1$. In such a case, all cells go through the same length S schedule, just delayed with respect to one another. When the condition produces an integer, then different cells can have different schedules. The number of schedules for different mutually exclusive groups of cells with the same schedules within each group, but different from group to group, is the number of values of m for which (11.30) is satisfied that are not integer multiples of previous values of m that solve the condition in (11.30). but if there are s such schedules of distinct lengths S_i , then

$$\sum_{i=1}^s S_i = \frac{\Delta}{2} \cdot L \quad (11.31)$$

See problem 11.6 for an interesting development of the above equations.

In the minimum-memory implementation described above, the relationship to the triangular structure in the memory connection is still inherent, but it evolves in time to prevent essentially idle memory cells.

Time variation of depth accompanying a data rate change

This triangular-interleaver interpretation in Figures 11.6, 11.8, and 11.9 (with or without minimum memory requirement) allows graceful change in operation of the interleaver depth between values that

maintain L and J co-prime and are both lower and or equal to the upper bound² of $J \leq M \cdot L + 1$. The overall delay must be held constant in time to do so. This description calls the new depth J' and the old depth J . There will be a point in time, call it time 0 at the beginning of a period where all new bytes entering the interleaver will observe depth J' while all bytes already in the interleaver (or de-interleaver) will observe the old depth J . The corresponding symbol periods will similarly be denoted by T and T' . (At a constant delay through interleaving and de-interleaver, the symbol rate necessarily must change, which implies a data-rate change.) To maintain the same delay in absolute time, and noting that impulse protection for such a delay remains the same, if the codeword length $N = L$ and correction capability remain the same (just the depth changes to accommodate the same length burst of noise/errors), then

$$T' = \frac{J-1}{J'-1} T \quad , \quad (11.32)$$

another advantage of using the symbol-clock (rather than interleaver period) notation and interpretation. Since bytes through the interleaver/de-interleaver combination must undergo exact the same delay whether before or after the depth change, a byte exiting the interleaver or de-interleaver will be “clocked out” at exactly the same point in absolute time. The time-slot-interchange positions also are at the same time. However, there are two symbol clocks in which to interpret absolute time. A master implementation clock is determined by the greatest common multiple, GCM, of $J-1$ and $J'-1$ as

$$\frac{1}{\delta T} = \frac{GCM}{J-1} \frac{1}{T} = \frac{GCM}{J'-1} \frac{1}{T'} \quad . \quad (11.33)$$

Essentially the higher speed clock will always touch in absolute time all write/read instants of the interleaver for either the old or the new depth. New bytes (symbols) enter the interleaver as shown (the interleaver has the same isosceles triangle structure in concept at any depth) in terms of L clock cycles of the new symbol clock with symbol period $T' = \frac{GCM}{J'-1} \frac{1}{\delta T}$ – and any memory element with a new byte after time zero in it will pass symbols $k \cdot (J'-1)T' = k \cdot GCM \cdot \delta T$ time slots later, where k is the index within the constant period of the interleaver. Old bytes within the interleaver exit on the $i(k) = k \cdot \Delta$ line at time instants $kT = k \cdot \frac{GCM}{J-1} \delta T$, while new bytes within the interleaver exit on the $i(k) = k \cdot \Delta'$ line at time instants $kT' = k \cdot \frac{GCM}{J'-1} \delta T$. All these time instants correspond to integer multiples of the high-rate clock. If at any time, both clocks are active then a read operation must be executed before the write operation on that clock cycle occurs (along with all the shifts in the proper order to not drop data within the delay elements shown if indeed the implementation was directly in the isosceles-triangular structure).

When no old bytes exist within the structure any longer (which is exactly the delay of the interleaver/de-interleaver combination), then the higher rate clock can be dropped and the new clock and new depth then used until any other depth change occurs in subsequent operation.

The time-variable structure uses the triangular structure to facilitate the explanation. This structure uses the constant $L(L-1)/2$ locations of memory independent of depth. However, the lower-memory requirement of the cell structure image discussed earlier can again be used even in the depth-variation case. The interleaver situation with larger depth will require a larger number of cells, so the number of cells increases with an increase in depth and decreases with a decrease in depth. For an increase in depth, $(J' - J) \frac{L-1}{2}$ more cells are needed in the interleaver (and the same number less for a decrease in depth). These cells can be allocated (or removed) as they are first necessary (or unnecessary).

Just as in the constant-depth case, an off-line algorithm (software) can be executed to determine the order for each new byte position of cell use by simply watching timers set according to the clock $\frac{1}{\delta T}$ as they sound. Any timer sounding at a time that is only a read can be reserved for potential future use. Any new byte to be written can use the cells reserved for use (either because it was vacated earlier by an isolated read at the old clock instants or because it is an additional cell needed). After $\frac{1}{2}(J-1)L(L-1)$ seconds, the interleaver algorithm (and de-interleaver algorithms) will have completely switched to the new depth. With such minimum-memory-cell use thus established, the depth change can then begin and follow the consequent pattern.

²The constraint can always be met with the generalized triangular interleaver by using a sufficiently large M and dummy symbols to maintain the co-prime depth and period.

11.2.4 Random Binary Interleaving

In random interleaving, the idea is to eliminate the regular patterns in $G(D)$ or the associated interleaving rule $\pi(k)$ over a very long period. Random interleaving is often used in turbo coding, as in Section 11.3. The **uniform random interleaver** is an abstraction that is really an average over many statistical possibilities for an interleaver with large period L . The idea is that for any pattern of l positions (the positions may be viewed as the location of 1's), it is equally likely to be interleaved into any of the

$$L_l = \binom{L}{l} \quad (11.34)$$

possible pattern of l positions. Clearly any interleaver will be deterministic and so could only approximate such an effect. However, over an ensemble of interleavers (essentially making the interleaver cyclostationary), the uniform random interleaver can be hypothesized. Such an interleaver does not necessarily guarantee a fixed depth and $J-1$ spaces between previously adjacent symbols. Those earlier “with depth” interleavers address burst noises. The uniform random interleaver instead is a concept used in trellis coding for AWGNs.

The main objective of random interleaving is to create a very long block length for the concatenated code when viewed as a single code. Codes selected randomly, as long as the block length is very long, often can achieve capacity (Section 8.3). Random interleaving tries to install this element of randomness in the code design without increasing complexity, presuming the use of iterative decoding of the two interleaved codes. The number of ways in which to make a specific type of error is essentially reduced by the large codeword/interleaver length and the unlikely possibility that a pattern of errors for a low-distance error event will just happen to touch the right places for both codes. Section 11.3 investigates turbo codes where the uniform random interleaver is presumed in analysis.

A random uniform interleaver then would translate any pattern of l ones into one of those patterns with probability, L_l^{-1} . This subsection lists 2 types of random interleavers that attempt to approximate uniform random interleaving:

Berrou- Glavieux Block Interleavers

The period $L = K \cdot J = 2^i \cdot 2^j$ is a power of 2 and eight prime numbers are used:

m	1	2	3	4	5	6	7	8
p_m	17	37	19	29	41	23	13	7

The time index within a block is $k = 0, \dots, L-1$. Recall that $(\cdot)_M$ means the quantity in brackets modulo M , i.e., the part left over after subtracting the largest contained integer multiple of M . Defining $r_0 = (k)_J$, $c_0 = (k - r_0)/J$, and $m = (r_0 + c_0)_8$ the interleaver order rule is

$$\pi(k) = c(k) + J \cdot r(k) \quad (11.35)$$

where

$$r(k) = (p_{m+1} \cdot (c_0 + 1) - 1)_K \quad (11.36)$$

and

$$c(k) = \left(\left(\frac{K}{2} + 1 \right) \cdot (r_0 + c_0) \right)_J. \quad (11.37)$$

This interleaver has a long period for reasonable values of K and J and causes a robust randomness of the positions of error events in one code with respect to another. An event that results from exceeding free/minimum distance in one constituent code is very unlikely to also be in just the right places after interleaving to exceed the free/minimum distance of another code. Thus the number of free/minimum distance events is greatly reduced, and at lower SNR when many error events in a first decoder are like to occur, this redistribution of error events dominates \bar{P}_b . As SNR increases, the unlikely nature of noise exceeding free/minimum distance begins to dominate. One could expect good performance/gain at lower SNR, and then the usual constituent code performance at higher SNR. Such was the insight of Berrou in developing turbo codes (see Section 11.3).

JPL (Jet Propulsion Laboratory) Block Interleaver

For any K and even J , define:

m	1	2	3	4	5	6	7	8
p_m	31	37	43	47	53	59	61	67

Again the time index is $k = 0, \dots, L - 1$. Defining $r_0 = \left(\frac{i-m}{2} - c_0\right)_J$, $c_0 = \left(\frac{i-m}{2}\right)_J$, and $m = (r_0)_8$ the interleaver order rule is

$$\pi(k) = 2 \cdot r(k) \cdot K \cdot c(k) - (k)_2 + 1 \quad (11.38)$$

where

$$r(k) = (19 \cdot r_0)_{\frac{K}{2}} \quad (11.39)$$

and

$$c(k) = (p_{m+1} \cdot c_0 + 21 \cdot (k)_2)_J \quad (11.40)$$

Pseudorandom Interleavers

Pseudorandom interleavers make use of pseudorandom binary sequences (PRBS). Such sequences are based on a theory of maximum-length polynomials. A PRBS circuit is a rate one convolutional code with constant 0 input and with feedback based on a polynomial $p(D)$ with implementation $G(D) = \frac{1}{p(D)}$. The degree- ν polynomial is chosen so that with binary arithmetic, it has no nontrivial factors (i.e., it is “prime”) and has other properties not discussed here.

Such circuits (if initialized with a nonzero initial condition) will generate a periodic sequence of period $2^\nu - 1$ that thus necessarily must include every nonzero binary pattern of length ν bits. The period of the interleaver can be no greater than the period of the PRBS. Table 11.2.4 lists such maximum-length polynomials³:

The pseudorandom interleaver uses such a PRBS to specify the output position of each input symbol. Thus, each successive output bit of the PRBS in combination with the last $\nu - 1$ such bits specifies an address $\pi(k)$ for the interleaver. If an address exceeds the interleaver period (when $L < 2^\nu - 1$), then it is discarded before use, and the PRBS circuit is cycled again. The de-interleaver can regenerate the same sequence and then successively extract the $(\pi(k))^{th}$ symbol and restore it to position k . Clearly, such pseudorandom interleaving corresponds to block interleaving unless the period is exactly $L = 2^\nu - 1$, in which case, the structure may have alternative implementations with less memory and delay.

S-Random Interleavers

An S-random interleaver tries to ensure that adjacent symbols are spaced further than S (and integer) after interleaving and to approximate also the random re-distribution of the uniform random interleaver. The S-random interleaver is designed by first choosing an $S \leq \sqrt{\frac{L}{2}}$ and running the algorithm in Figure 11.10. This algorithm will converge if the condition on S is met and the second integer S' is ignored. Usually designers run the algorithm several times increasing S' until they find the largest such value for which the algorithm converges. The correlation between intrinsic and extrinsic information for the S-random interleaver decays exponential with the difference between interleaver indices (and has the exception value zero for time difference of 0).

³These were taken from the book Error Control Systems by Wickert (used in EE387 at Stanford), which has a far more complete listing of all the possible polynomials for each ν (there are many). See Chapter 12 of our text for implementations of scramblers. Note that the polynomials in x in that reference (as well as most coding books) corresponds to an advance, not a delay, which is why the reversal of the polynomials in Table 11.2.4

ν	$P(D)$
2	$1 + D + D^2$
3	$1 + D^2 + D^3$
4	$1 + D^3 + D^4$
5	$1 + D^3 + D^5$
6	$1 + D^5 + D^6$
7	$1 + D^6 + D^7$
8	$1 + D^4 + D^5 + D^6 + D^8$
9	$1 + D^5 + D^9$
10	$1 + D^7 + D^{10}$
11	$1 + D^9 + D^{11}$
12	$1 + D^6 + D^8 + D^{11} + D^{12}$
13	$1 + D^9 + D^{10} + D^{12} + D^{13}$
14	$1 + D^9 + D^{11} + D^{13} + D^{14}$
15	$1 + D + D^{15}$
16	$1 + D + D^3 + D^{12} + D^{16}$
17	$1 + D^3 + D^{17}$
18	$1 + D^7 + D^{18}$
23	$1 + D^5 + D^{23}$
24	$1 + D^{17} + D^{22} + D^{23} + D^{24}$
31	$1 + D^{28} + D^{31}$

Table 11.5: A list of maximum-length (primitive) polynomials.

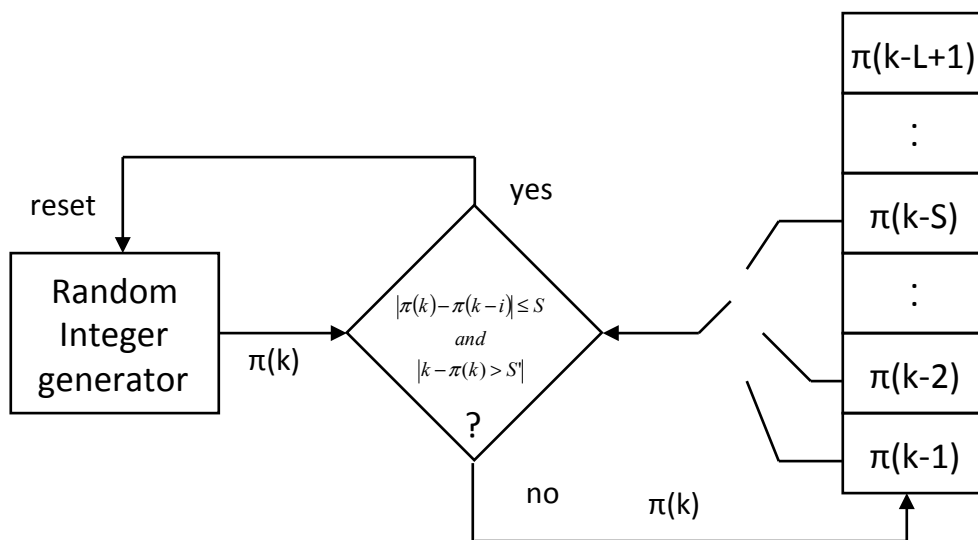


Figure 11.10: S-random interleaver illustrated.

11.3 Turbo Codes

Turbo Codes are parallel or serial concatenations of simple good convolutional codes with significant interleaving first discovered by Claude Berrou in 1993. Interleaving is used with turbo codes and essentially the additional gain accrues to a reduction in the nearest neighbor count by the depth of the interleaver. Turbo Code design targets a data rate that is less than capacity, but perhaps just slightly less. Subsection 11.3.1 investigates the rate of simple concatenations, while Subsection 11.3.2 introduces the concept of code puncturing, which is very useful in implementing the desired \bar{b} for a turbo code. Probability-of-error approximations appear in Subsections 11.3.3 and 11.3.4 for parallel and serial concatenations respectively. Subsections 11.3.5 and 11.3.6 enumerate various convolutional codes that have been found to also be good for turbo-code use when $\bar{b} < 1$.

11.3.1 Turbo-Code rate definition

Definition 11.3.1 (Turbo Code) *A Turbo Code is a parallel or serial concatenation of two convolutional codes with uniform random interleaving (or an approximation to it) to distribute the error events of the two codes with respect to one another. Turbo codes are designed with the expectation of iterative (i.e., “turbo”) decoding’s use at the receiver.*

parallel concatenations

The rate of a parallel concatenation of two systematic convolutional codes (where the information bits are sent only once, along with the parity bits of both codes) is again

$$\frac{1}{\bar{b}} = \frac{1}{\bar{b}_1} + \frac{1}{\bar{b}_2} - 1 \quad . \quad (11.41)$$

The -1 term is eliminated if the codes are not systematic.⁴

A few examples illustrate the possibilities:

EXAMPLE 11.3.1 (basic rate 1/3) If two convolutional codes both have rate $\bar{b}_1 = \bar{b}_2 = .5$, then $\bar{b} = 1/3$.

EXAMPLE 11.3.2 (rate 1/6) If two different convolutional codes both have rates $\bar{b}_1 = 1/3$ and $\bar{b}_2 = 1/4$, then $\bar{b} = 1/6$.

EXAMPLE 11.3.3 (higher rates) Higher rate turbo codes can be constructed from higher-rate convolutional codes. If two convolutional codes both have rate $\bar{b}_1 = \bar{b}_2 = 3/4$, then $\bar{b} = 3/5$. If two convolutional codes both have rate $\bar{b}_1 = \bar{b}_2 = .8$, then $\bar{b} = 2/3$.

It may be difficult to find a convolutional code with high \bar{b} and good distance properties without a large number of states, so the puncturing of Subsection 11.3.2 becomes the preferred alternative to implementation of the high-rate turbo codes enumerated in Subsection 11.3.5.

serial concatenations

Serial turbo codes have rate equal to the product of the constituent code rates

$$\bar{b} = \bar{b}_1 \cdot \bar{b}_2 \quad . \quad (11.42)$$

A serial turbo code constructed from two rate 1/2 codes would have rate 1/4. Similarly a serial turbo code from two rate 2/3 codes has rate 4/9 (or less than 1/2). Clearly serial concatenation requires very high rates for the two used codes if the concatenation is to be high rate. The puncturing of the Subsection 11.3.2 is helpful for reducing the rate loss in serial concatenation of binary codes.

⁴This formula only applies if all the codes have less than one bit per dimension.

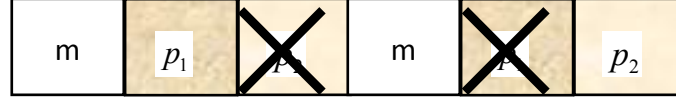


Figure 11.11: Puncturing of rate 1/3 turbo (or convolutional) code to rate 1/2.

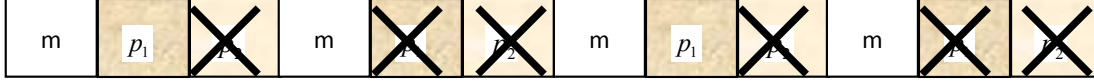


Figure 11.12: Puncturing of rate 1/3 turbo (or convolutional) code to rate 2/3.

11.3.2 Puncturing

Before proceeding to Turbo Codes, this subsection first introduces puncturing of a convolutional code; a concept often used in either or both of the convolutional codes in a concatenated Turbo Coding system. Puncturing is the regular/periodic deletion of parity bits from a convolutional encoder output to increase the rate of the code. Such deletion can reduce the free distance of the code. Puncturing allows a single code to be used at different \bar{b} , which may be useful in systems that transmit at different data rates depending on conditions. Design or use of a different code for each data rate might complicate implementation. With turbo codes, it is the interleaver that provides the additional gain and it may be that the overall code sees little difference with puncturing from what might have been achieved using a better constituent high-rate code. Thus, puncturing is often used with turbo codes to simplify the implementation at several rates with the same encoder.

For instance, Figure 11.11 illustrates how the rate 1/3 turbo code that results from parallel concatenation of two rate-1/2 convolutional codes can be restored to rate-1/2 by alternately deleting one of the two parity bits. Some performance loss with respect to rate-1/3 turbo coding might be expected, but of course at an increase in data rate. Often, the resultant higher-rate code is still a very good code for the new higher data rate.

A yet higher data rate can be achieved as in Figure 11.12: A frame of 12 bits, 4 information and 8 parity, retains only 2 of the parity bits to increase the data rate to 2/3. Similarly, a rate 2/3 turbo, based on two rate 2/3 convolutional codes, could alternately delete 1 of the 2 parity bits generated at each symbol period.

Generally, a systematic code with k information bits per symbol and $n - k$ parity bits per symbol can be punctured to a higher rate q/p by accepting kq input bits and deleting $nq - kp$ of the parity bits,

$$\frac{kq}{nq - (nq - kp)} = \frac{q}{p} . \quad (11.43)$$

The deleted bits in turbo coding may not be the same for each successive symbol to “distribute” the loss of parity, so j successive symbols may be used. In this most general case, the puncturing can be described by the $nqj \times kpj$ singular permutation/generator matrix G_{punc} . For the puncturing in Figure 11.11 with $j = 2$, $k = 1$, $n = 3$, $q = 1$ and $p = 2$

$$G_{punc} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} , \quad (11.44)$$

accepting 6 bits but outputting 4. The punctured code words are generated by multiplying the output code sequence from the original code by the generator, or $\mathbf{v}_{punc}(D) = \mathbf{v}(D) \cdot G_{punc}$ where $\mathbf{v}(D)$ may be written in terms of “stacking” j successive symbols. Puncturing can be very useful to avoid excessive constellation expansion when applying turbo codes to PAM/QAM transmission systems with $\bar{b} \geq .5$.

11.3.3 Analysis of probability of bit error for parallel concatenation

The analysis of Turbo Code probability of error makes use of the presumption of the uniform random interleaver in Section 11.2. This subsection begins with the example of the well-known 4-state rate-1/2 code.

A systematic encoder realization is

$$G_1(D) = \left[1 \quad \frac{1 + D + D^2}{1 + D^2} \right] . \quad (11.45)$$

The two columns of the parity matrix (or effectively the two output bits of the encoder) have been reversed in order with respect to nominal use of this well-known 4-state convolutional code. Clearly such an output-bit-order reversal changes no property of the code itself (although the number of input bit errors and encoder mappings are different). Any input bit stream with weight $w = 1$ (or only one 1 in the stream so $u(D) = D^m$ where m is some integer) will lead to an infinite string of output 1's corresponding to infinite weight⁵, which means that for this encoder the likelihood of a single input bit error occurring is essentially zero. In a parallel concatenated Turbo Code constructed from two uses of this code, inputs with 2 bit errors are thus of more interest. A length-2 input sequence, however, of $1 + D^2$ produces an output sequence of $v(D) = [1 + D^2 \quad 1 + D + D^2]$ and thus produces the minimum-distance (5) error-event code sequence. The parallel concatenation then would make an error if this two-bit input error event were to occur on one of the codes and also after de-interleaving to correspond to $D^m \cdot (1 + D^2)$ for any of $m = 0, \dots, L - 1$ within the period L of the interleaver. For the uniform random interleaver, this simultaneous event has probability of occurring

$$\left(\frac{L}{2} \right)^{-1} = \frac{2}{L(L-1)} , \quad (11.46)$$

for a particular value of m and thus probability $L \cdot \frac{2}{L(L-1)} = \frac{2}{L-1}$ that it could occur for any of the L values $m = 0, \dots, L - 1$. Furthermore, the rate-1/3 concatenated code has a $d_{free} = 8 = 2 + 3 + 3$ (and corresponds to essentially $v_{turbo}(D) = [1 + D^2 \quad 1 + D + D^2 \quad \pi(1 + D + D^2)]$ where π has been used loosely to denote that the ordering of the 3 parity bits from the second interleaved use of the code occurs in 3 positions that are not the same times as the first 3 parity bits. The coding gain of the overall concatenated code is $8/3 = 2.67 = 4.26$ dB. The original “mother” code had gain $5/2 = 2.5 = 3.97$ dB and thus differs only by .3 dB⁶. However, the nearest neighbor coefficient for this minimum-distance event in the Turbo-Code concatenation is smaller. Since two input bit errors occur when half the minimum distance of the code has been exceeded by noise projection on the error event, then the probability that the $d_{free} = 8$ error event occurs for this turbo code is then

$$\bar{P}_b(d_{free}) \approx \frac{2}{L-1} b \cdot a(d_{free}, b) \cdot Q(\sqrt{d_{free} \cdot \text{SNR}}) \quad (11.47)$$

$$\approx \frac{4}{L-1} \cdot 1 \cdot Q(\sqrt{8 \cdot \text{SNR}}) , \quad (11.48)$$

and corresponds to 2 input bit errors. There are no error events for the convolutional code that correspond to 3 (or any odd number greater than 3) of input bit errors (but there are error events corresponding to 4 and higher even numbers of input bit errors). Table 11.6 lists the error event combinations and shows the coefficient for interleaver depth $L = 101$. Table 11.7 repeats some measured results from a popular author. The error coefficients are also listed for periods of 1,001 and 10,001. The simulation

⁵An infinite number of channel errors must occur for only 1 input bit error.

⁶Assuming as is usual in convolutional codes that extra dimensions can be added as in method 1 of Section 10.1.

distance	in-ev code 1	in-ev code 2	$2 \cdot a(d, 2)$	d_{free} construct	$\frac{4 \cdot a(d, 2)}{100}$	$\frac{4 \cdot a(d, 2)}{1000}$	$\frac{4 \cdot a(d, 2)}{10000}$
8	$1 + D^2$	$D^m \cdot (1 + D^2)$	2	2+3+3	.04	.004	.004
9	$1 + D^2$ $1 + D^4$	$D^m \cdot (1 + D^4)$ $D^m \cdot (1 + D^2)$	4	2+3+4 2+4+3	.08	.08	.008
10	$1 + D^2$ $1 + D^4$ $1 + D^6$	$D^m \cdot (1 + D^6)$ $D^m \cdot (1 + D^4)$ $D^m \cdot (1 + D^2)$	6	2+3+5 2+4+4 2+5+3	.12	.012	.0012
11	$1 + D^2$ $1 + D^4$ $1 + D^6$ $1 + D^8$	$D^m \cdot (1 + D^8)$ $D^m \cdot (1 + D^6)$ $D^m \cdot (1 + D^4)$ $D^m \cdot (1 + D^2)$	8	2+3+6 2+4+5 2+5+4 2+6+3	.16	.016	.0016
12	$1 + D^2$ $1 + D^4$ $1 + D^6$ $1 + D^8$ $1 + D^{10}$	$D^m \cdot (1 + D^{10})$ $D^m \cdot (1 + D^8)$ $D^m \cdot (1 + D^6)$ $D^m \cdot (1 + D^4)$ $D^m \cdot (1 + D^2)$	10	2+3+7 2+4+6 2+5+5 2+6+4 2+7+3	.20	.02	.002
t	$1 + D^2$ \vdots $1 + D^{2(t-8)}$	$D^m \cdot (1 + D^{2(t-8)})$ \vdots $D^m \cdot (1 + D^2)$	$2(t-7)$	\vdots	$.04(t-7)$	$.004(t-7)$	$.0004(t-7)$

Table 11.6: Enumeration of $b = 2$ input bit error events and corresponding codeword error event construction for $G_1(D) = (1 + D + D^2)/(1 + D^2)$.

distance	N_b for $L = 10^2$	N_b for $L = 10^3$	N_b for $L = 10^4$
8	3.89×10^{-2}	3.99×10^{-3}	3.99×10^{-4}
9	7.66×10^{-2}	7.96×10^{-3}	7.99×10^{-4}
10	.1136	1.1918×10^{-2}	1.1991×10^{-3}
11	.1508	1.5861×10^{-2}	1.5985×10^{-3}
12	.1986	1.9887×10^{-2}	1.9987×10^{-3}
13	.2756	2.4188×10^{-2}	2.4017×10^{-3}
14	.4079	2.9048×10^{-2}	2.8102×10^{-3}
15	.6292	3.4846×10^{-2}	3.2281×10^{-3}
16	1.197	6.5768×10^{-2}	6.0575×10^{-3}

Table 11.7: 4-state convolutional code as turbo constituent

results are very close, except that when the length of the error event becomes large relative to the period, then some accuracy is lost (because Table 11.6 essentially ignored the finite period in enumerating error-event pairs).

It is interesting to view the same example with the output bit order reversed to the usual instance of this code so that

$$G_2(D) = \left[1 \quad \frac{1 + D^2}{1 + D + D^2} \right] . \quad (11.49)$$

This encoder maps a weight 5 codeword error event of $[1 + D + D^2 \quad 1 + D^2]$ to 3 input bit errors. With concatenation, then this error event corresponds to $d_{free} = 7 = 3 + 2 + 2$ indicating that the concatenated code does not have the same codewords and indeed has a lower minimum distance. Nonetheless, this minimum distance requires 3 input-bit positions of errors to coincide for the two codes, which has probability (with uniform random interleaving) of

$$L \cdot \binom{L}{3}^{-1} = \frac{6}{(L-1)(L-2)} . \quad (11.50)$$

distance	in-ev code 1	in-ev code 2	$2 \cdot a(d, 2)$	d_{free} construct	$\frac{4 \cdot a(d, 2)}{100}$
10	$1 + D^3$	$D^m \cdot (1 + D^3)$	2	2+4+4	.04
12	$1 + D^3$	$D^m \cdot (1 + D^6)$	4	2+4+6	.08
	$1 + D^6$	$D^m \cdot (1 + D^3)$		2+6+4	
14	$1 + D^3$	$D^m \cdot (1 + D^9)$	6	2+4+8	.12
	$1 + D^6$	$D^m \cdot (1 + D^6)$		2+6+6	
	$1 + D^9$	$D^m \cdot (1 + D^3)$		2+8+4	
16	$1 + D^3$	$D^m \cdot (1 + D^{12})$	8	2+4+10	.16
	$1 + D^6$	$D^m \cdot (1 + D^9)$		2+6+8	
	$1 + D^9$	$D^m \cdot (1 + D^6)$		2+8+6	
	$1 + D^{12}$	$D^m \cdot (1 + D^3)$		2+10+4	
t	$1 + D^3$	$D^m \cdot (1 + D^{1.5(t-8)})$	$t-8$	\vdots	.02($t-8$)
	\vdots $1 + D^{1.5(t-8)}$	\vdots $D^m \cdot (1 + D^3)$			

Table 11.8: Enumeration of $b = 2$ input bit error events and corresponding codeword error event construction for $G_2(D) = (1 + D^2)/(1 + D + D^2)$.

Thus, 3-bit error events have very small probability, even if they correspond to minimum-distance codeword events (unless the SNR is very high so that d_{\min} dominates all nearest-neighbor-like coefficients). In fact, the weight two input error event with smallest codeword distance is $1 + D^3$ and corresponds to an output distance of $d = 10$. Repeating Table 11.6 now in Table 11.8 yields (there are no odd-number free distances for 2-input-bit-error events).

The second instance of the concatenated code with $G_2(D)$ is clearly better because the distances are larger, and the error coefficients are the same. (The improvement is roughly 1 dB). However, at high SNRs, the dominant free-distance-7 error event will eventually make asymptotic performance worse. So, up to some SNR, the second code is better and then it is worse after that SNR. The eventual dominance of minimum distance at high SNR causes the contribution of different error events to dominate, leading to an unusual flattening of the \bar{P}_b curve known as the “error floor” in turbo coding. In both concatenated codes, the larger free distance from the rate 1/3 code is only a slight improvement. The main improvement is the reduction in bit-error probability caused by division by the interleaver period. Such an improvement is often called **interleaver gain**. Over the range of normal use of codes, a factor of 10 usually corresponds to 1 dB improvement, so the interleaver gain for this code would then be

$$\gamma_{interleaver} = \log_{10}((L - 1)/2) \quad (11.51)$$

at least up to a few dB (at which point the approximation of 1 dB per factor of 10 is no longer valid).

In a more general situation of a rate $1/n$ code

$$G_{sys}(D) = \left(1 \quad \frac{g_1(D)}{g_0(D)} \quad \dots \quad \frac{g_{n-1}(D)}{g_0(D)} \right) \quad , \quad (11.52)$$

and any weight-one input information sequence $u(D) = D^m$ produces an infinite-weight output codeword (or large finite weight if the code is terminated in a finite packet length). Thus, again only weight 2 (or higher) input errors are of interest since any $g_0(D)$ must divide $1 + D^j$ for some sufficiently large j . Since input-bit weight 3 errors have a coefficient factor of $\frac{6}{(L-1)(L-2)}$, then they typically have much lower contribution to \bar{P}_b unless the SNR is sufficiently high and the distance for these errors is smaller.

Error Flooring Error flooring occurs when the SNR is sufficiently high that the smaller nearest-neighbor coefficient for error events with 2 errors (or more) is overwhelmed by the exponential decrease with SNR of the Q-function at high SNR. At such high SNR, d_{\min} error events will again dominate. Figure 11.13 depicts this effect generically for two types of error events for which one has a $d_2 = 5$ (output distance corresponding to 2-input bit errors), but has a larger coefficient of .01, while the second

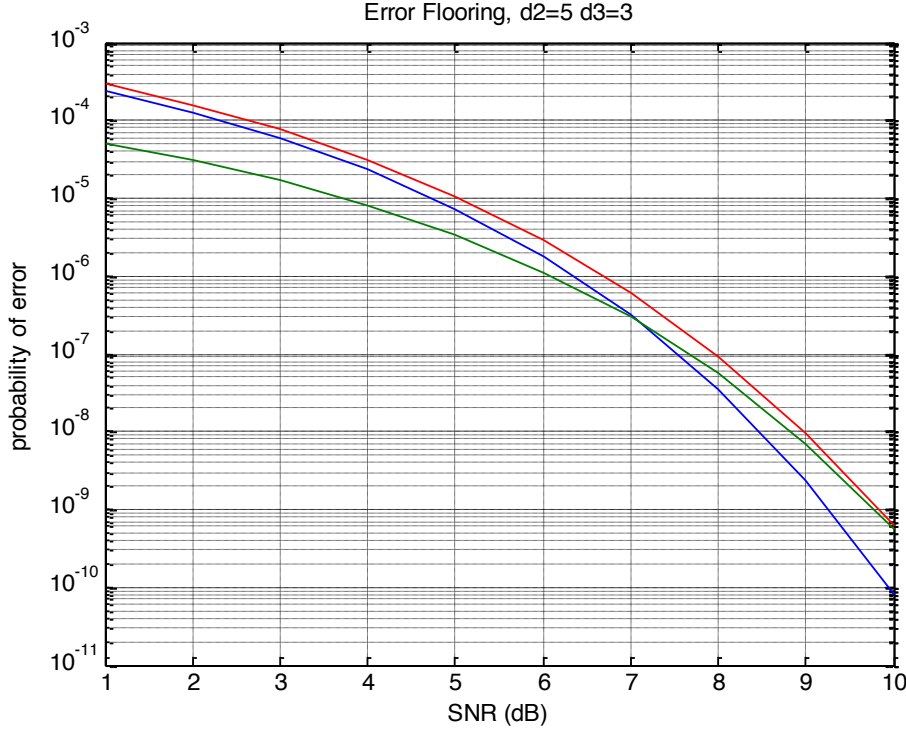


Figure 11.13: Generic depiction of basic error flooring in Turbo Codes.

has a smaller $d_3 = 3$ (output distance corresponding to 3 input bit errors), but a coefficient of .001. At smaller SNR, the low coefficient of the Q-function term with smaller distance causes this term to be negligible, but eventually the exponential decay of the Q-function term with larger distance causes it instead to be negligible. In the middle of Figure 11.13 where the overall error probability (the upper curve, which is sum of the two curves) deviates temporarily from a pure exponential decay and “curves less or flattens” temporarily until resuming the exponential decay again for the lower distance portion. Thus, the smaller error-coefficient term dominates at low SNR and provides lower total probability of error. Thus, operation with a Turbo Code attempts to choose parameters so that this portion of the curve corresponds to the range of interest. Over this \bar{P}_b range, if 2-input-bit error events dominate, then

$$\bar{P}_{b,turbo} \approx \frac{2\bar{N}_b(d_2)}{L} Q\left(\frac{d_{\min}}{2\sigma}\right) = \frac{4}{L} Q\left(\frac{d_{2,cat}}{2\sigma}\right), \quad (11.53)$$

where $d_{2,cat}$ is the 2-input-bit-error codeword distance for the concatenated code system. The probability of bit error thus reduces by the factor $L/2$. More generally with rate k/n codes, the situation is more complicated because a simple 2-bit error pattern “cancelling all the denominators” is more complicated and the interleaver possibilities multiply. Aversion of this conceptual complexity uses puncturing of rate $1/n$ codes rather than direct design for k/n codes.

Figure 11.14 illustrates the probability of error for interleaver depths of $L = 100$, $L = 1000$, and $L = 10000$ for the 4-state turbo rate 1/2 Turbo Code (with encoder G_1) of Table 11.9 with puncturing. From the code itself, one expects a gain of 4.7 dB plus another 3.7 dB for $L = 10000$, so then roughly 8.4 dB also. The actual plotted gain is slightly less at about 8 dB because of the effects and contributions of higher-distance terms.

Figure 11.15 illustrates the convergence of the same code as in Figure 11.14 for the interleaver size of 1000. The probability of symbol error (and thus the probability also of bit error for this code) converges within just a few iterations. Figure 11.16 compares the probability of error for SOVA and APP, while the difference is larger at very low SNR, the difference becomes small in the range of operation of $\bar{P}_b = 10^{-6}$.

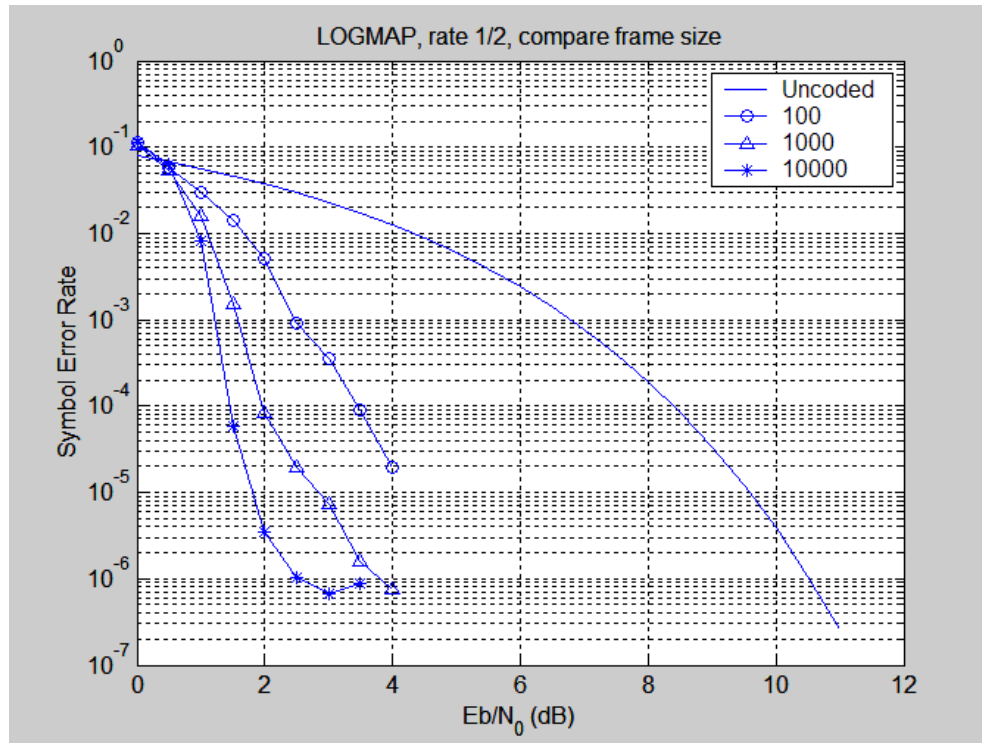


Figure 11.14: Probability of error for the rate-1/2 4-state turbo code, with puncturing to rate 1/2.

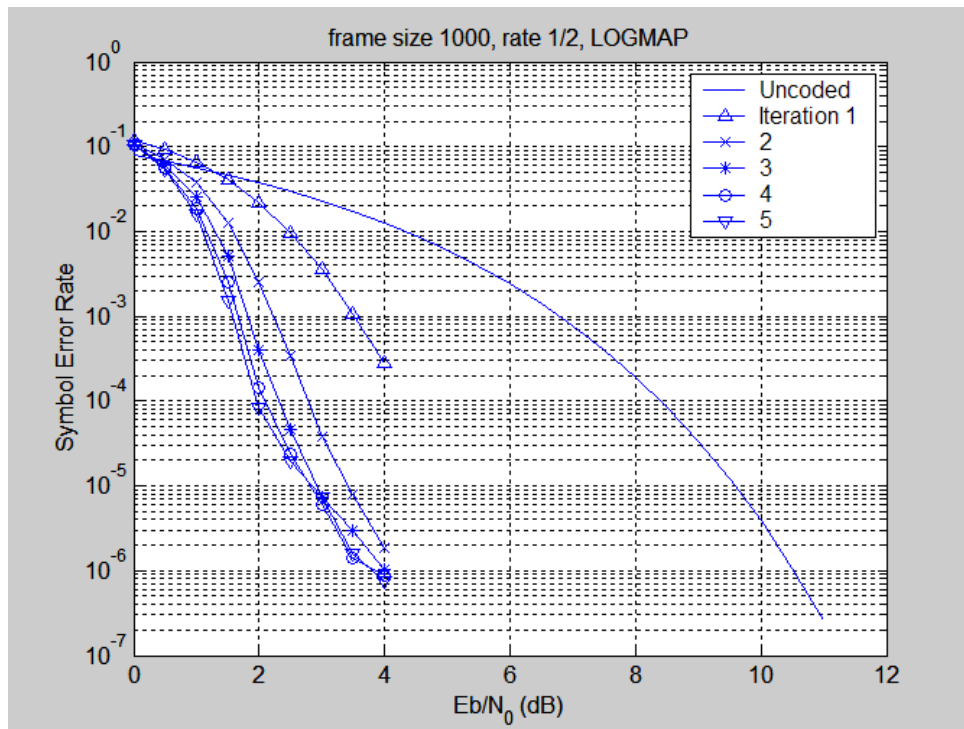


Figure 11.15: APP Convergence of Probability of error for the rate-1/2 4-state turbo code, with puncturing to rate 1/2.

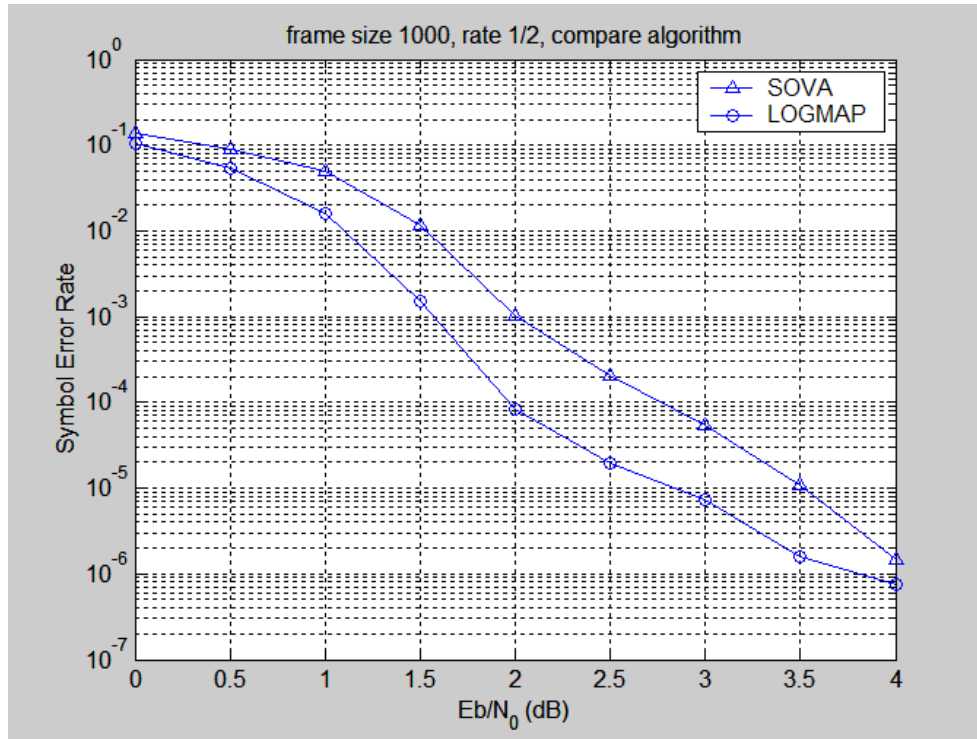


Figure 11.16: Comparison of probability of error for SOVA and APP on the rate 1/2 (punctured) turbo code.

Figure 11.17 illustrates the additional gain of not puncturing and using rate 1/3 instead, which is about .7 dB at $\bar{P}_b = 10^{-6}$ for this example. Actually, slightly larger because this rate 1/3 code also sees a larger d distribution generally than the punctured rate 1/2 code. This gain is small for this code, as is typical with puncturing in turbo codes that little is lost in terms of coding gain.

11.3.4 Analysis of probability of bit error for serial concatenation

Serial concatenation of codes has a somewhat different analysis than parallel concatenation. Analysis of serial concatenation retains the basic concept that an error event with several bit errors is less likely to fall with random interleaving into the exact positions that cause errors in two codes simultaneously. However, the re-encoding into new additional parity bits of the parity (or all) bits of the inner code causes some changes in the probability of error formulas. Often in serial turbo-code concatenation, the two codes are different and the ensuing analysis accommodates such difference.

First, the error coefficient simply reflects the fact that in serial concatenation the probability that $\left\lceil \frac{d_{free}^{out}}{2} \right\rceil$ bits after uniform random interleaving of depth L again fall in “all the wrong places in the outer code (out)” is

$$\left(\frac{L}{\left\lceil \frac{d_{free}^{out}}{2} \right\rceil} \right) . \quad (11.54)$$

This expression is often approximated in serial turbo-code analysis by

$$C \cdot L^{-\left(\left\lceil \frac{d_{free}^{out}}{2} \right\rceil\right)} , \quad (11.55)$$

where the exponential dependence on the codeword (or encoder output) free distance rather than the number of input bit errors, distinguishes serial concatenation from parallel concatenation in Equations

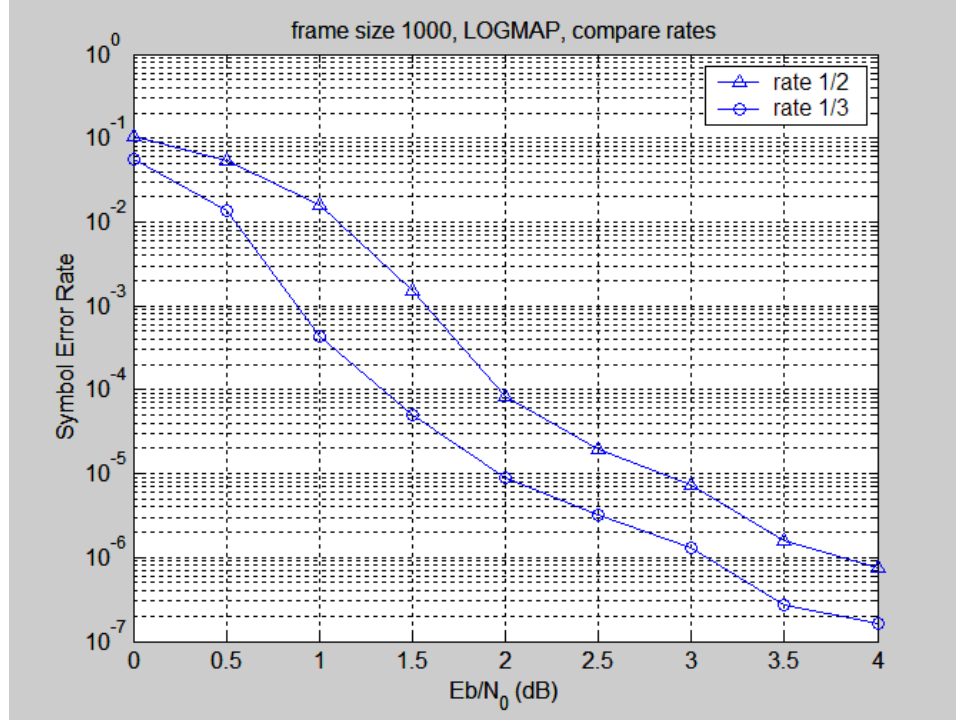


Figure 11.17: Comparison of rate 1/3 and rate 1/2 with puncturing for 4-state Turbo code.

11.46 and 11.50. C is a constant. This coefficient multiplies the error coefficient for the outer Turbo Code, whatever that coefficient is, for both symbol-error and bit-error probability expressions. While this factor follows parallel concatenation, the adjustment to the Q-function argument requires additional scrutiny for serial concatenation. This text assumes that at least the inner encoder is always systematic so that at least two-input-bit errors are necessary to cause it to have finite output distance and thus to have non-zero probability of the error event. This inner distance for 2-bit errors is called d_2^{in} and similarly the 3-bit error distance is called d_3^{in} . If the inner code's decoder has had the error event occur, then that inner decoder's error probability has Q-function argument $\sqrt{d_2^{in} \cdot \text{SNR}}$. These 2 input bits (which are also part of the output distance in a systematic realization) are "already guaranteed" to be contributing to the output error event for the outer code so the outer decoder's tolerance for errors is reduced to⁷

$$\left\lceil \frac{d_{free}^{out} - 3}{2} \right\rceil \quad (11.56)$$

The overall effect on the Q-function argument for an overall decoder is

$$\left\lceil \left[\frac{d_{free}^{out} - 3}{2} \right] \cdot d_2^{in} + d_w^{in} \right\rceil \cdot \text{SNR} \quad (11.57)$$

where $d_w^{in} = d_2^{in}$ for situations in which the outer code has even d_{free}^{out} and $d_w^{in} = d_3^{in}$ for odd d_{free}^{out} . The reader should recall that the SNR in (11.57) scales down as the product of the two code rates, which is $\bar{b} = \bar{b}_{in} \cdot \bar{b}_{out}$. Thus, the additional error-correcting power (or distance) of the outer code applies to all the extra bits that must additionally be in error in the outer code and thus multiplies d_2^{in} but does multiply

⁷The use of the greatest integer function in (11.56) allows the argument to be reduced by 3 when outer free distance is odd and by only 2 when outer free distance is 2 in agreement with the reality that codes with odd distances essentially get one more position of possible error before experiencing an error event.

the common bits. However those common bits do have to be in error and whence the last additive term of d_w^{in} in (11.57). The total-bit-error-counting quantity

$$N_b(d) = \sum_{b=1}^{\infty} b \cdot a(d, b) \quad (11.58)$$

has special use in serial concatenation. With it, the overall probability of bit error expression is then

$$\begin{aligned} \bar{P}_b &\approx L \left(\left\lceil \frac{L}{\left\lceil \frac{d_{free}^{out}}{2} \right\rceil} \right\rceil \right)^{-1} \cdot \frac{N_b(d_{free}^{in}) \cdot N_b(d_{free}^{out})}{b} \cdot Q \left(\sqrt{\left\lceil \frac{d_{free}^{out} - 3}{2} \right\rceil \cdot d_2^{in} + d_w^{in}} \cdot \text{SNR} \right) \quad (11.59) \\ &= L \left(\left\lceil \frac{L}{\left\lceil \frac{d_{free}^{out}}{2} \right\rceil} \right\rceil \right)^{-1} \cdot \frac{N_b(d_{free}^{in}) \cdot N_b(d_{free}^{out})}{b} \cdot Q \left(\sqrt{2 \cdot \left\lceil \frac{d_{free}^{out} - 3}{2} \right\rceil \cdot d_2^{in} + d_w^{in}} \cdot \bar{b} \cdot \frac{\mathcal{E}_b}{\mathcal{N}_0} \right) \quad (11.60) \end{aligned}$$

where the second expression uses the “energy per bit” form. It is clear that an outer code with odd free distance is preferable in terms of Q-function argument per complexity. However, the outer encoder need not be systematic nor even use feedback. The interleaving gain thus is slightly altered with respect to parallel concatenation to be

$$\gamma_{serial} = \log_{10} \left(\frac{L!}{\left\lceil \frac{d_{free}^{out}}{2} \right\rceil!} \right) \text{ dB} \quad (11.61)$$

over the range of operation of 10^{-4} to 10^{-7} . This factor can be large for $d_{free}^{out} > 2$, but the product of $N_b(d)$ terms reduces the effect of the gain. Error flooring in serial concatenation follows the same basic principle as in parallel concatenation, namely that eventually as SNR increase, minimum distance dominates probability of error and thus an event with larger free distance than d_2^{in} or d_w^{in} but also smaller error coefficient could eventually be expected to be the major contributor to probability of error.

11.3.5 Coding Tables for Parallel Concatenation with rate $1/n$, or for rate $n - 1/n$ with no puncturing

The following tables are extracted from some papers by Divsalar at the course website. Some of the data in the tables has been augmented by this text’s author to include more information pertinent to code implementation and analysis. All the codes appear to be the best or among the best known for the various complexities and parameters listed.

The 4-state rate-1/2 convolutional code that has been previously studied has a systematic realization $[1(1 + D^2)/(1 + D + D^2)]$. The resultant rate-1/3 turbo code can be punctured alternately on the parity bits of the two encoders in each successive symbol with uniform random interleaving on the information bits. The left-most values of d_2 , d_3 , and d_{free} are for the base or mother code itself. The right-most values are for the turbo code or concatenated code and thus can be used directly in probability of error analysis expressions like (11.53). The rule for adding the 3 right-most additional columns is to find the difference between the d_{free} and d_i listed for the base code, subtract 2 or 3 for d_2 or d_3 respectively, and then add that amount to the distance. $d_{free,cat}$ is the new minimum of the two quantities $d_{2,cat}$ and $d_{3,cat}$. Thus, $d_{2,cat} = 2 \cdot d_2 - 2$ and $d_{3,cat} = 2 \cdot d_3 - 3$ and $d_{free,cat} = \min\{d_{2,cat}, d_{3,cat}\}$. With puncturing, the puncturing pattern of 101010 was used for the rate 1/2 code in the worst position in terms of weight for the base code.

Table 11.9 lists the distances for 2 and 3 input bit errors as well as d_{free} for the overall-rate 1/2 turbo code by Divsilar (the overall code values were determined by this author and omitted by Divsilar)” Another more complete set of rate-1/2 punctured codes will appear shortly in Subsection 11.3.6.

The best rate 1/3 codes found by Divsilar (again with overall turbo code values found by this author) appear in Table 11.10:

The best rate 1/4 codes found by Divsilar (again with overall turbo code values found by this author) appear in Table 11.11:

2^ν	$g_0(D)$	$g_1(D)$	d_2	d_3	d_{free}	$d_{2,cat}$	$d_{3,cat}$	$d_{free,cat}$
4	7	5	4	3	3	6	5	5
8	13	15	5	4	4	8	6	6
16	23	37	7	4	4	10	8	8

Table 11.9: Rate 1/2 Constituent PUNCTURED convolutional codes for turbo codes

2^ν	$g_0(D)$	$g_1(D)$	$g_2(D)$	d_2	d_3	d_{free}	$d_{2,cat}$	$d_{3,cat}$	$d_{free,cat}$
2	3	2	1	4	∞	4	6	∞	6
4	7	5	3	8	7	7	14	11	11
8	13	17	15	14	10	10	26	17	17
16	23	33	37	22	12	12	42	21	21

Table 11.10: Rate 1/3 Constituent convolutional codes for turbo codes

The best rate 2/3 codes found by Divsilar (again with overall turbo code values found by this author) appear in Table 11.12:

The best rate 3/4 codes found by Divsilar (again with overall turbo code values found by this author) appear in Table 11.13:

11.3.6 Parallel and Serial Turbo Code Tables with puncturing for base rate 1/2

To enumerate best codes with puncturing, Deanesgharan, Laddomada and Mondin (DLM) have searched over all rate 1/2 codes and puncturing patterns for different rates to find best parallel and serial concatenations for up to 32 states. The mother codes used in the puncturing search are to create high rate $(n-1)/n$ codes after puncturing. These codes are listed for 4 and 8 states in Table 11.14 and for 16 and 32 states in Table 11.15. An indication of SNR means among all encoders with the same d_2 , the one with minimum SNR to get $\bar{P}_b = 10^{-6}$ was selected. The entry d_3 means instead that the code with maximum d_3 was selected. The entry d_2 is a code with largest d_2 .

Parallel concatenations with the same code Table 11.16 lists the best parallel-concatenation codes (presuming the same code is used twice). The codes are rate $(n-1)/n$ after puncturing. The puncturing pattern is octal and corresponds to the mother code rate-1/2 output bit pairs (info, parity) are enumerated from left to right in increasing time and the puncturing pattern is pressed on top with 0 meaning puncture that parity bit. For instance 5352 means 101 011 101 010 so and corresponds to (letting i_k be an information bit and p_k be the corresponding parity bit)

$$(i_1, p_1, i_2, p_2, i_3, p_3, i_4, p_4, i_5, p_5, i_6, p_6) \rightarrow (i_1, i_2, i_3, p_3, i_4, i_5, i_6) \quad . \quad (11.62)$$

Puncturing patterns need not always maintain a systematic code, although it is rare in parallel concatenation to see puncturing of information bits. The distances shown are for the mother code itself. As earlier, the overall $d_{free,cat}$ for performance analysis of the concatenated system is found as $d_{free,cat} = \min_{i=2,3} 2 \cdot d_i - i$.

Serial concatenations - inner codes Table 11.17 lists best known inner codes with puncturing patterns generated from rate 1/2 mother codes. Often the puncturing can delete information bits (meaning the parity bit carries better information under puncturing than the information bit itself). Puncturing is applied to the inner code output and the rate is for the resultant punctured inner code and is $(n-1)/n$.

2^v	$g_0(D)$	$g_1(D)$	$g_2(D)$	$g_3(D)$	d_2	d_3	d_{free}	$d_{2,cat}$	$d_{3,cat}$	$d_{free,cat}$
8	13	17	15	11	20	12	12	38	21	21
16	23	35	27	37	32	16	14	62	31	31

Table 11.11: Rate 1/4 Constituent convolutional for turbo codes

2^v	$h_0(D)$	$h_1(D)$	$h_2(D)$	d_2	d_3	d_{free}	$d_{2,cat}$	$d_{3,cat}$	$d_{free,cat}$
4	7	3	5	4	3	3	6	3	3
8	13	15	17	5	4	4	8	5	5
16	23	35	27	8	5	5	14	7	7
16	45	43	61	12	6	6	22	9	9

Table 11.12: Rate 2/3 Constituent convolutional codes for turbo codes

Serial concatenations - outer codes Table 11.18 lists best known outer codes with puncturing patterns generated from rate 1/2 mother codes. Often the puncturing can delete information bits (meaning the parity bit carries better information under puncturing than the information bit itself). Puncturing is applied to the outer code output and the rate is for the resultant punctured outer code and is $(n-1)/n$.

A serial turbo-code design would choose one code from Table 11.17 and one from 11.18, using (11.60) to evaluate the performance and the overall code rate is $\bar{b} = \bar{b}_{in} \cdot \bar{b}_{out}$.

SubsectionCDMA 2000 Turbo Code

To be added at a later date. See document at web page for more information.

2^ν	$h_0(D)$	$h_1(D)$	$h_2(D)$	$h_3(D)$	d_2	d_3	d_{free}	d_{free}	$d_{2,cat}$	$d_{3,cat}$	$d_{free,cat}$
4	7	5	3	1	3	3		3	4	3	3
8	13	15	17	11	4	4		4	6	5	5
16	23	35	33	25	5	4		4	8	5	5

Table 11.13: Rate 3/4 Constituent convolutional codes for turbo codes

2^ν	$\begin{bmatrix} 1 & \frac{g_1}{g_0} \end{bmatrix}$	d	N_e	$N_b(d)$	d_2	d_3
4 (d_2)	$\begin{bmatrix} 1 & \frac{5}{7} \end{bmatrix}$	5	1	3	6	5
		6	2	6		
		7	4	14		
		8	8	32		
		9	16	72		
4 (SNR)	$\begin{bmatrix} 1 & \frac{7}{5} \end{bmatrix}$	5	1	2	5	∞
		6	2	6		
		7	4	14		
		8	8	32		
		9	16	72		
8 (d_2)	$\begin{bmatrix} 1 & \frac{15}{13} \end{bmatrix}$	6	2	6	8	6
		8	10	40		
		10	49	245		
		12	241	1446		
		14	1185	8295		
8 (d_2)	$\begin{bmatrix} 1 & \frac{17}{13} \end{bmatrix}$	6	1	4	8	7
		7	3	9		
		8	5	20		
		9	11	51		
		10	25	124		
8 (SNR)	$\begin{bmatrix} 1 & \frac{15}{17} \end{bmatrix}$	6	1	2	6	∞
		7	3	12		
		8	5	20		
		9	11	48		
		10	25	126		

Table 11.14: Best 4- and 8-sate Rate 1/2 Constituent (mother) convolutional codes for for use with puncturing in turbo codes

2^ν	$\left[1 \frac{g_1}{g_0} \right]$	d	N_e	$N_b(d)$	d_2	d_3
16 (d_2)	$\left[1 \frac{33}{31} \right]$	7	2	8	12	7
		8	4	16		
		9	6	26		
		10	15	76		
		11	37	201		
16	$\left[1 \frac{21}{37} \right]$	6	1	2	6	∞
		7	1	5		
		8	3	10		
		9	5	25		
		10	12	56		
16 (d_2)	$\left[1 \frac{27}{31} \right]$	7	2	8	12	7
		8	3	12		
		9	4	16		
		10	16	84		
		11	37	213		
16 (d_2)	$\left[1 \frac{37}{23} \right]$	6	1	4	12	8
		8	6	23		
		10	34	171		
		12	174	1055		
		14	930	6570		
16 (d_2)	$\left[1 \frac{33}{23} \right]$	7	2	8	12	7
		8	4	16		
		9	6	26		
		10	15	76		
		11	37	201		
16 (d_2)	$\left[1 \frac{35}{23} \right]$	7	2	8	12	7
		8	3	12		
		9	4	16		
		10	16	84		
		11	37	213		
16 (SNR)	$\left[1 \frac{23}{35} \right]$	7	2	6	7	∞
		8	3	12		
		9	4	20		
		10	16	76		
		11	137	194		
32 (d_3)	$\left[1 \frac{71}{53} \right]$	8	3	12	12	∞
		10	16	84		
		12	68	406		
		14	860	6516		
		16	3812	30620		
32 (SNR d_2)	$\left[1 \frac{67}{51} \right]$	8	2	7	20	8
		10	20	110		
		12	68	406		
		14	469	3364		
		16	2560	20864		

Table 11.15: Best 16- and 32-state Rate 1/2 Constituent (mother) convolutional codes for use with puncturing in turbo codes

$n - 1$	4 states	8 states	16 states	32 states
2	$\left[1 \frac{5}{7} \right]$ 13 (3,1,3) $d_2 = 4, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 13 (4,3,10) $d_2 = 5, d_3 = 4$	$\left[1 \frac{37}{23} \right]$ 13 (4,2,6) $d_2 = 7, d_3 = 4$	$\left[1 \frac{67}{51} \right]$ 13 (5,2,7) $d_2 = 9, d_3 = 5$
3	$\left[1 \frac{5}{7} \right]$ 56 (3,4,10) $d_2 = 3, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 53 (3,2,5) $d_2 = 3, d_3 = 3$	$\left[1 \frac{37}{23} \right]$ 53 (3,1,3) $d_2 = 4, d_3 = 3$	$\left[1 \frac{67}{51} \right]$ 53 (4,2,7) $d_2 = 7, d_3 = 4$
4	$\left[1 \frac{5}{7} \right]$ 253 (2,1,2) $d_2 = 2, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 253 (3,9,24) $d_2 = 3, d_3 = 3$	$\left[1 \frac{37}{23} \right]$ 253 (3,3,9) $d_2 = 4, d_3 = 3$	$\left[1 \frac{67}{51} \right]$ 253 (3,1,3) $d_2 = 5, d_3 = 3$
5	$\left[1 \frac{5}{7} \right]$ 1253 (2,2,4) $d_2 = 2, d_3 = 3$	$\left[1 \frac{17}{13} \right]$ 1253 (3,15,40) $d_2 = 3, d_3 = 3$	$\left[1 \frac{27}{31} \right]$ 1272 (3,2,6) $d_2 = 4, d_3 = 3$	$\left[1 \frac{71}{53} \right]$ 1272 (4,108,406) $d_2 = 4, d_3 = \infty$
6	$\left[1 \frac{5}{7} \right]$ 5352 (2,22,44) $d_2 = 2, d_3 = 3$	$\left[1 \frac{17}{13} \right]$ 5253 (2,1,2) $d_2 = 2, d_3 = 3$	$\left[1 \frac{27}{31} \right]$ 5253 (3,12,33) $d_2 = 3, d_3 = 3$	$\left[1 \frac{71}{53} \right]$ 5253 (3,3,6) $d_2 = 3, d_3 = \infty$
7	$\left[1 \frac{5}{7} \right]$ 25253 (2,7,14) $d_2 = 2, d_3 = 3$	$\left[1 \frac{15}{17} \right]$ 25253 (2,7,14) $d_2 = 2, d_3 = \infty$	$\left[1 \frac{33}{23} \right]$ 25253 (2,1,2) $d_2 = 2, d_3 = 3$	$\left[1 \frac{67}{51} \right]$ 25253 (2,1,2) $d_2 = 3, d_3 = 3$
8	$\left[1 \frac{5}{7} \right]$ 125253 (2,9,18) $d_2 = 2, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 125253 (2,4,8) $d_2 = 2, d_3 = 3$	$\left[1 \frac{37}{23} \right]$ 125253 (2,1,2) $d_2 = 2, d_3 = 3$	$\left[1 \frac{67}{51} \right]$ 125253 (3,17,49) $d_2 = 3, d_3 = 3$

Table 11.16: Best puncturing patterns for given high-rate parallel turbo codes

$n - 1$	4 states	8 states	16 states	32 states
2	$\left[1 \frac{5}{7} \right]$ 7 (3,1,3) $d_2 = 5, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 7 (4,3,10) $d_2 = 7, d_3 = 4$	$\left[1 \frac{27}{31} \right]$ 7 (4,1,5) $d_2 = 11, d_3 = 5$	$\left[1 \frac{67}{51} \right]$ 7 (5,2,7) $d_2 = 19, d_3 = 6$
3	$\left[1 \frac{5}{7} \right]$ 27 (2,1,4) $d_2 = 4, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 27 (3,2,9) $d_2 = 6, d_3 = 3$	$\left[1 \frac{27}{31} \right]$ 27 (3,1,5) $d_2 = 10, d_3 = 4$	$\left[1 \frac{67}{51} \right]$ 65 (4,7,51) $d_2 = 18, d_3 = 5$
4	$\left[1 \frac{5}{7} \right]$ 67 (2,4,13) $d_2 = 4, d_3 = 3$	$\left[1 \frac{15}{13} \right]$ 127 (3,9,52) $d_2 = 6, d_3 = 3$	$\left[1 \frac{27}{31} \right]$ 351 (4,16,176) $d_2 = 10, d_3 = 4$	$\left[1 \frac{67}{51} \right]$ 325 (4,22,191) $d_2 = 18, d_3 = 5$
5	$\left[1 \frac{5}{7} \right]$ 527 (2,6,26) $d_2 = 4, d_3 = 2$	$\left[1 \frac{15}{13} \right]$ 527 (2,1,6) $d_2 = 6, d_3 = 3$	$\left[1 \frac{27}{31} \right]$ 635 (3,8,43) $d_2 = 10, d_3 = 4$	$\left[1 \frac{67}{51} \right]$ 1525 (3,3,21) $d_2 = 18, d_3 = 5$
6	$\left[1 \frac{5}{7} \right]$ 3525 (2,84,2693) $d_2 = 4, d_3 = 2$	$\left[1 \frac{17}{13} \right]$ 2527 (2,4,20) $d_2 = 6, d_3 = 4$	$\left[1 \frac{37}{23} \right]$ 6525 (2,4,22) $d_2 = 10, d_3 = 5$	$\left[1 \frac{67}{51} \right]$ 6525 (3,4,34) $d_2 = 18, d_3 = 5$
7	$\left[1 \frac{5}{7} \right]$ 12527 (2,15,74) $d_2 = 4, d_3 = 2$	$\left[1 \frac{17}{13} \right]$ 12527 (2,7,42) $d_2 = 6, d_3 = 4$	$\left[1 \frac{33}{23} \right]$ 32525 (2,6,51) $d_2 = 10, d_3 = 4$	$\left[1 \frac{67}{51} \right]$ 32525 (3,14,135) $d_2 = 18, d_3 = 5$
8	$\left[1 \frac{5}{7} \right]$ 72525 (2,153,5216) $d_2 = 4, d_3 = 2$	$\left[1 \frac{15}{13} \right]$ 52527 (2,4,32) $d_2 = 6, d_3 = 3$	$\left[1 \frac{33}{23} \right]$ 72525 (2,6,42) $d_2 = 10, d_3 = 4$	$\left[1 \frac{67}{51} \right]$ 152525 (3,23,299) $d_2 = 18, d_3 = 5$

Table 11.17: Best puncturing patterns for given high-rate serial (inner code) turbo codes

$n - 1$	4 states	8 states	16 states	32 states
2	$\left[1 \frac{7}{5} \right]$ 15 (3,1,2) $d_2 = 3, d_3 = \infty$	$\left[1 \frac{15}{13} \right]$ 13 (4,3,10) $d_2 = 5, d_3 = 4$	$\left[1 \frac{33}{31} \right]$ 13 (5,7,25) $d_2 = 6, d_3 = 5$	$\left[1 \frac{71}{53} \right]$ 13 (6,15,60) $d_2 = 6, d_3 = \infty$
3	$\left[1 \frac{5}{7} \right]$ 56 (3,4,10) $d_2 = 2, d_3 = 3$	$\left[1 \frac{15}{17} \right]$ 33 (4,29,126) $d_2 = 4, d_3 = \infty$	$\left[1 \frac{23}{35} \right]$ 17 (4,29,150) $d_2 = 4, d_3 = \infty$	$\left[1 \frac{71}{53} \right]$ 36 (4,1,4) $d_2 = 8, d_3 = \infty$
4	$\left[1 \frac{5}{7} \right]$ 253 (2,1,2) $d_2 = 2, d_3 = 3$	$\left[1 \frac{15}{17} \right]$ 136 (3,5,16) $d_2 = 3, d_3 = \infty$	$\left[1 \frac{27}{31} \right]$ 351 (4,16,176) $d_2 = 10, d_3 = 4$	$\left[1 \frac{71}{53} \right]$ 133 (4,28,192) $d_2 = 8, d_3 = \infty$
5	$\left[1 \frac{5}{7} \right]$ 1253 (2,2,4) $d_2 = 2, d_3 = 3$	$\left[1 \frac{17}{13} \right]$ 1253 (3,15,40) $d_2 = 3, d_3 = 3$	$\left[1 \frac{33}{31} \right]$ 653 (4,98,436) $d_2 = 4, d_3 = 4$	$\left[1 \frac{71}{53} \right]$ 1272 (4,108,406) $d_2 = 4, d_3 = \infty$
6	$\left[1 \frac{7}{5} \right]$ 3247 (2,4,8) $d_2 = 2, d_3 = \infty$	$\left[1 \frac{17}{13} \right]$ 5253 (2,1,2) $d_2 = 2, d_3 = 3$	$\left[1 \frac{27}{31} \right]$ 3352 (3,7,24) $d_2 = 4, d_3 = 3$	$\left[1 \frac{71}{53} \right]$ 5253 (3,3,6) $d_2 = 3, d_3 = \infty$
7	$\left[1 \frac{7}{5} \right]$ 15247 (2,6,12) $d_2 = 2, d_3 = \infty$	$\left[1 \frac{15}{17} \right]$ 15652 (2,2,4) $d_2 = 2, d_3 = \infty$	$\left[1 \frac{23}{35} \right]$ 13632 (3,13,52) $d_2 = 3, d_3 = \infty$	$\left[1 \frac{71}{53} \right]$ 13172 (3,4,18) $d_2 = 5, d_3 = \infty$
8	$\left[1 \frac{7}{5} \right]$ 65247 (2,9,18) $d_2 = 2, d_3 = \infty$	$\left[1 \frac{15}{17} \right]$ 65256 (2,3,6) $d_2 = 2, d_3 = \infty$	$\left[1 \frac{33}{31} \right]$ 123255 (3,21,75) $d_2 = 4, d_3 = 3$	$\left[1 \frac{67}{51} \right]$ 124672 (3,11,36) $d_2 = 5, d_3 = 3$

Table 11.18: Best puncturing patterns for given high-rate serial (outer code) turbo codes

code	m_1	m_2	m_3	m_4
parity1	1	-	1	-
parity2	-	1	-	1

Table 11.19: $b = 1$ with 4QAM, or $\bar{b} = .5$

11.4 Turbo Codes for Higher-Level Constellations

The turbo codes of Section 11.3 can be used effectively with any constellation by simply using log likelihood ratios to compute the initial information from the constellation and otherwise executing iterative decoding between 2 (or more) instances of the code. However, while the codes are very good and have high gain, the gain for multi-level constellations may not be consistent as the number of points in the constellation M grows. Thus, researchers have investigated some codes to be used with S-random interleavers that will provide a relatively constant gap for $\bar{b} \geq 0.5$. This section presents two of those codes, an 8-state code found by Sadjadpour, Sonalkar, and Calderbank (SSC) when working at ATT on DMT DSL systems, where the constant gain is useful for loading algorithm implementation and a 16-state code found by Pons, Sorbara, and Duvaut (PSD), who were also working on DSL. The SSC code has a coding gain (with interleaver of about 2000 bits) of 6.5 dB and thus a gap of 3 dB at 10^{-7} probability of error. The 16-state PSD has about 6.8 dB coding gain alone with a similar length interleaver and was observed when used with an outer hard-decoding that was not involved in iterative decoding to have a gain of about 7.1 dB, and thus a gap of only 2.4 dB. Given that shaping gain would contribute another 1.53 dB, these codes are essentially within a 1-2 dB of capacity or absolute best performance for multi-level transmission systems.

11.4.1 The 8-state SSC Code

The code has the already identified rate-1/2 encoder $G(D) = [1 \frac{17}{13}]$, which thus confirms that this binary convolutional code is a good one to use. This code is used twice. The real innovations of SSC are in the puncturing patterns and actually a few tenths of dB in a clever augmentation to the S-random interleaver search. There are actually two interleavers specified on each of the input branches to the use of this code. One of the interleavers acts on ONLY the systematic bits before they are transmitted, while the other rearranges those bits on the input to the second parity-computing instance of the same code.

Those extra steps over the algorithm in Figure 11.10 are compute the autocorrelation matrix between extrinsic LLR's and information bits, trying to drive these on average to as low of values as possible. The procedure is in the 3rd ATT paper on the web site. The second information-bit-transmitted-only interleaver is a deterministic design that is also described in the paper at the web site.

The length of the interleaver chosen for examples by SSC is $L = 2176$ bits, but the procedure for interleaver design can be repeated for many situations and data rates, thus consequently different interleaver lengths. Such an interleaver length is relatively small for the large gains evident, thus keep interleaver delay in implementations to reasonable levels of a few thousand bits (typically a few ms or less for modern data rates of a few Mbps or more).

The code uses different puncturing patterns for different \bar{b} . Tables 11.19-11.26 list the value of \bar{b} ; the constellation used with coding, with each dimension mapped with gray code (no two adjacent points differ by more than 1 bit in label in a single dimension, see Figure 11.18 for an example); and up to 15 bits per 2-dimensional symbol, along with parity bits if used. A 1 indicates the parity bit is transmitted, while a - indicates the parity bit is punctured for the rate 1/2 code. Vertical lines in the table indicate symbol boundaries with respect to the information bits. Thus, the code in Table 11.19 could be viewed as 8-dimensional, while the codes in Tables 11.20, 11.21, 11.23, and 11.25 could be viewed as 4-dimensional.

code	m_1	m_2	m_3	m_4
parity1	1	-	1	-
parity2	-	1	-	1

Table 11.20: $b = 2$ with 16QAM, or $\bar{b} = 1$

code	m_1	m_2	m_3	m_4	m_5	m_6
parity1	1	-	-	-	-	-
parity2	-	-	-	1	-	-

Table 11.21: $b = 3$ with 16QAM, or $\bar{b} = 1.5$

code	m_1	m_2	m_3	m_4
parity1	1	-	-	-
parity2	-	-	1	-

Table 11.22: $b = 4$ with 64QAM, or $\bar{b} = 2$

code	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
parity1	1	-	-	-	1	-	-	1	-	-
parity2	-	-	1	-	-	1	-	-	-	1

Table 11.23: $b = 5$ with 256QAM, or $\bar{b} = 2.5$

code	m_1	m_2	m_3	m_4	m_5	m_6
parity1	1	-	-	-	-	-
parity2	-	-	-	1	-	-

Table 11.24: $b = 6$ with 256QAM, or $\bar{b} = 3$

code	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}
parity1	1	-	-	-	-	-	1	-	-	-	1	-	-	-
parity2	-	-	-	1	-	-	-	1	-	-	-	-	-	1

Table 11.25: $b = 7$ with 1024QAM, or $\bar{b} = 3.5$

code	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
parity1	1	-	-	-	-	-	-	-
parity2	-	-	-	-	1	-	-	-

Table 11.26: $b = 8$ with 1024QAM, or $\bar{b} = 4$

code	m_1	m_2 through m_4	m_5	m_6 through m_8	m_9
parity1	1	-	-	-	1
parity2	-	-	1	-	-

Table 11.27: $b = 9$ with 4096QAM, or $\bar{b} = 4.5$ - flip parity 1 and 2 on alternate symbols

code	m_1	m_2 through m_9	m_{10}
parity1	1	-	-
parity2	-	-	1

Table 11.28: $b = 10$ with 4096QAM, or $\bar{b} = 5$

code	m_1	m_2 through m_5	m_6	m_7 through m_{10}	m_{11}
parity1	1	-	-	-	1
parity2	-	-	1	-	-

Table 11.29: $b = 11$ with 16384QAM, or $\bar{b} = 5.5$ - flip parity 1 and 2 on alternate symbols

code	m_1	m_2 through m_{11}	m_{12}
parity1	1	-	-
parity2	-	-	1

Table 11.30: $b = 12$ with 16384QAM, or $\bar{b} = 6$

code	m_1	m_2 through m_6	m_7	m_8 through m_{12}	m_{13}
parity1	1	-	-	-	1
parity2	-	-	1	-	-

Table 11.31: $b = 13$ with 65536QAM, or $\bar{b} = 6.5$ - flip parity 1 and 2 on alternate symbols

code	m_1	m_2 through m_{11}	m_{12}
parity1	1	-	-
parity2	-	-	1

Table 11.32: $b = 14$ with 65536QAM, or $\bar{b} = 7$

code	m_1	m_2 through m_7	m_8	m_9 through m_{14}	m_{15}
parity1	1	-	-	-	1
parity2	-	-	1	-	-

Table 11.33: $b = 15$ with 262144QAM, or $\bar{b} = 7.5$ - flip parity 1 and 2 on alternate symbols

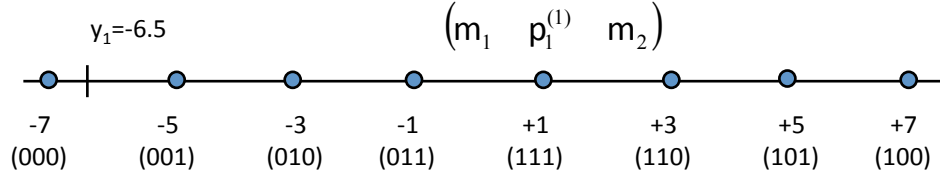


Figure 11.18: Example for feeding decoder.

Feeding the Decoder

EXAMPLE 11.4.1 (Decoder Feeding from one dimension of constellation) An example for one dimension of a 64 QAM constellation and rate 2/3 code illustrates decoding with the AT&T code. Figure 11.18 shows one dimension of the constellation, labelings, and a hypothesized received value of $y = -6.5$. The received value of -6.5 in the first dimension corresponds to the 3 bits $(m_1, p_1^{(1)}, m_2)$. The second parity bit is removed via puncturing. A trellis branch of interest in computation/input of γ_k in the APP of Section 9.3 would have the label $(m_1, p_1^{(1)}) = (1, 1)$ in the encoder. There are two constellation points that correspond to this combination, at +1 and at +3. Thus, gamma for this first code on this branch is

$$\gamma_k(1, 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \left(e^{-\frac{1}{2\sigma^2}(7.5)^2} + e^{-\frac{1}{2\sigma^2}(9.5)^2} \right) \quad (11.63)$$

The second γ_{k+1} affected by the same received symbol has a branch in the code trellis with label (0,1), but the second bit was punctured and not transmitted. Thus any point that has $m_2 = 0$ is a possible point for the sum and there are four such points for the first code

$$\gamma_{k+1}(0, 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \left(e^{-\frac{1}{2\sigma^2}(5)^2} + e^{-\frac{1}{2\sigma^2}(3.5)^2} + e^{-\frac{1}{2\sigma^2}(9.5)^2} + e^{-\frac{1}{2\sigma^2}(13.5)^2} \right) \quad (11.64)$$

For the second code, both parity bits are punctured. Thus, $\gamma_{k+1}(0, 1)$ remains the same as for the first code on the same branch, but now

$$\gamma_k(1, 1)(code2) = \frac{1}{\sqrt{2\pi\sigma^2}} \left(e^{-\frac{1}{2\sigma^2}(7.5)^2} + e^{-\frac{1}{2\sigma^2}(9.5)^2} + e^{-\frac{1}{2\sigma^2}(11.5)^2} + e^{-\frac{1}{2\sigma^2}(13.5)^2} \right) \quad (11.65)$$

11.4.2 The 16-state PSD Code

This code uses an S-random interleaver, which can be augmented by the same search as for the SSC code. The generator is $G(D) = \begin{bmatrix} 1 & 35 \\ & 23 \end{bmatrix}$, which is again among those found in the previous section as good for turbo codes with puncturing. This code adds a few tenths of a dB in coding gain for the doubling of decoder complexity and can otherwise use the same constellations and puncturing patterns as the SSC code. Most productive use would require searches of interleaving rules that corresponding to specific data rates to reduce correlation between extrinsic information and intrinsic information between decoders. In this case, the code appears to have coding gain of about 6.8 dB. An outer hard-decoder was found by PSD to add an additional .3 dB (with no additional interleaving), meaning a coding gain of 7.1 dB.

11.4.3 The Third Generation Wireless Turbo Code

Third Generation Wireless, specifically CDMA-2000 3GPP2, use a turbo code that is typically mapped into various forms of BPSK or QPSK signaling on the output bits of the encoder. Figure 11.20 illustrates the base encoder, which typically has punctured outputs. The encoder for both encoders is $G(D) = \begin{bmatrix} 1 & \frac{1+D+D^3}{1+D^2+D^3} & \frac{1+D+D^2+D^3}{1+D^2+D^3} \end{bmatrix}$, which is one of the codes in the earlier tables. The encoder outputs are

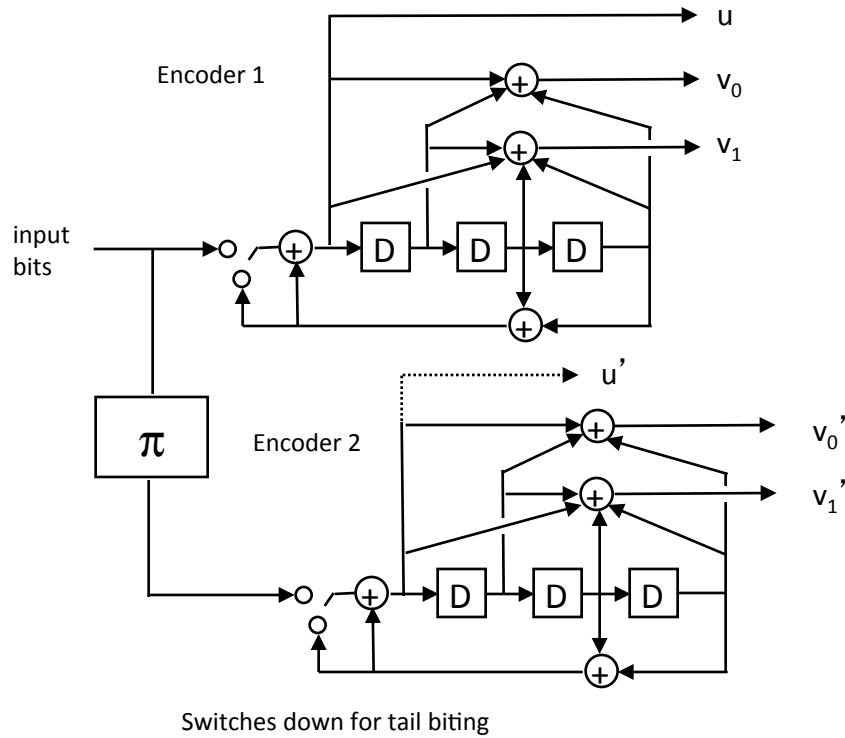


Figure 11.19: 3GPP Turbo encoder base (often punctured).

punc	1/2	1/3	1/4	1/5
u	11	11	11	11
v_0	10	11	11	11
v_1	00	00	10	11
v'_0	01	11	01	11
v'_1	00	00	11	11

Table 11.34: Puncturing patterns for various rates of the 3GPP Turbo Encoders

punctured to create a rate \bar{b} encoder according to Table 11.34 over two successive input bits (or symbol periods). This code is not intended for multi-level constellations and instead is an example of a good one used at low \bar{b} .

Various packet lengths are used for the code with tail biting that causes the switch to go to the lower position (the delay is all zeros at the beginning of each packet) with encoder 1's output bits first transmitted for each symbol followed by encoder 2's output bits. When the switch is in the lower position the feedback bit is transmitted. Since this bit is different for the 1st encoder than for the 2nd encoder, both are transmitted and thus the puncturing patterns are changed for the tail biting. Basically $\frac{6}{b}$ bits from the tail are transmitted after puncturing of the tail, or $\frac{3}{b}$ from each of the encoder outputs. The puncturing patterns for the tail biting appear in Table 11.35. The entries 2 and 3 mean to repeat the bit twice or three times respectively.

The interleaver for the 3GPP code approximates a random interleaver, but with a simplified implementation shown in Figure ???. This interleaver is characterized by an integer $n = 4, 5, \dots, 10$ and the packet sizes (without including tail-biting bits) in Table ??. A counter is initialized to zero at the beginning of the packet and its output is the input to the circuit in Figure ??. When a successful address (the packet length is listed as N_{turbo} in Figure ??), then the counter is incremented. This process is continued until all N_{turbo} input addresses have an interleave output address. The de-interleave process basically

punc	1/2		1/3		1/4		1/5	
u	111	000	222	000	222	000	333	000
v_0	111	000	111	000	111	000	111	000
v_1	000	000	000	000	111	000	111	000
u'	000	111	000	222	000	222	000	333
v'_0	000	111	000	111	000	1111	000	111
v'_1	000	000	000	000	000	111	000	111

Table 11.35: Puncturing patterns for tail biting of the 3GPP Turbo Encoders

n	packet length
378	4
570	5
762	5
1146	6
1530	6
2298	7
3066	7
4602	8
6138	8
9210	9
12282	9
20730	10

Table 11.36: Packet lengths and n for 3GPP turbo code.

reverses the diagram (and requires the solution of the least-significant-bit division problem, which is essentially a set of linear binary equations). The interleaver can also be stored in a large look-up table, but this 3GPP implementation is much simpler. The small look-up table is summarized in Table 11.21.

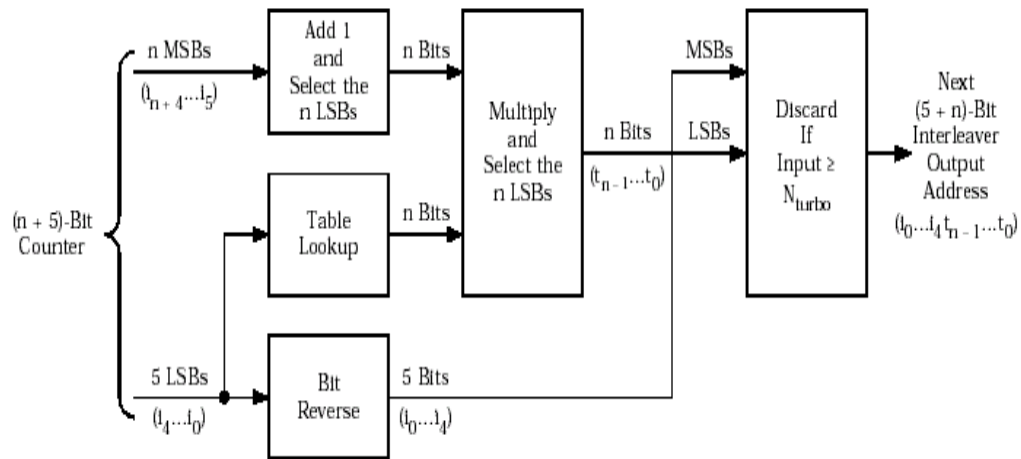


Figure 11.20: 3GPP Turbo encoder interleaver circuit.

Table 2.1.3.1.4.2.3-2. Turbo Interleaver Lookup Table Definition

Table Index	n = 4 Entries	n = 5 Entries	n = 6 Entries	n = 7 Entries	n = 8 Entries	n = 9 Entries	n = 10 Entries
0	5	27	3	15	3	13	1
1	15	3	27	127	1	335	349
2	5	1	15	89	5	87	303
3	15	15	13	1	83	15	721
4	1	13	29	31	19	15	973
5	9	17	5	15	179	1	703
6	9	23	1	61	19	333	761
7	15	13	31	47	99	11	327
8	13	9	3	127	23	13	453
9	15	3	9	17	1	1	95
10	7	15	15	119	3	121	241
11	11	3	31	15	13	155	187
12	15	13	17	57	13	1	497
13	3	1	5	123	3	175	909
14	15	13	39	95	17	421	769
15	5	29	1	5	1	5	349
16	13	21	19	85	63	509	71
17	15	19	27	17	131	215	557
18	9	1	15	55	17	47	197
19	3	3	13	57	131	425	499
20	1	29	45	15	211	295	409
21	3	17	5	41	173	229	259
22	15	25	33	93	231	427	335
23	1	29	15	87	171	83	253
24	13	9	13	63	23	409	677
25	1	13	9	15	147	287	717
26	9	23	15	13	243	193	313
27	15	13	31	15	213	57	757
28	11	13	17	81	189	501	189
29	3	1	5	57	51	313	15
30	15	13	15	31	15	489	75
31	5	13	33	69	67	391	163

Figure 11.21: 3GPP Turbo interleaver look-up table.

(t_c, t_r)	\bar{b}	deviation from capacity
(3,6)	.5	1.1 dB
(4,8)	.5	1.6 dB
(5,10)	.5	2.0 dB
(3,5)	.4	1.3 dB
(4,6)	1/3	1.4 dB

Table 11.37: Regular Low-Density Parity Code Average Performance.

11.5 Low-Density Parity Check Codes

Low-Density Parity Check (LDPC) codes were first studied by Gallager of MIT in the early 1960's. They essentially are an implementation of the concept of random coding, which is known from Chapter 8 to lead to code designs that can approach capacity. These block codes have very long block length N and a parity matrix $H(D) = H(0) = H$ that is essentially chosen randomly to have a number of 1's that grows linearly with block length (rather than as the square of the block length as might be expected if 0's and 1's were selected with equal probability for the H matrix entries). Also, from the expertise gained in Section 9.6, "cycles of 4" are avoided so that iterative decoding using the constraint-based structure of Section 9.6 can be used effectively.

Definition 11.5.1 (cycle of 4) *A cycle of 4 occurs in a code when for any 1's in the (i, j) and (i, k) positions in any row i of H , there is at least one other row $i' \neq i$ that also has 1's in the j^{th} and k^{th} positions.*

Definition 11.5.2 (Regular Parity Matrix) *A regular code with a regular parity matrix has exactly t_r 1's in every row and exactly t_c 1's in every column.*

The $(n - k) \times n$ parity matrix H is chosen in regular LDPC codes to have exactly t_c 1's in each column and exactly t_r 1's in each row, from which one notes

$$(n - k) \cdot t_r = n \cdot t_c \quad (11.66)$$

or equivalently

$$\bar{b} = 1 - \frac{t_c}{t_r} \quad (11.67)$$

Essentially, if uniformly distributed integers from 1 to n were selected, t_r at a time, they would represent the positions of the 1's in a row of H . Successive rows could be generated by selecting successively groups of t_r integers. If a row is obtained that is either linearly dependent on previous rows, or forms a 4-cycle (which one can determine is simply a rectangle of 4 1's in the corners somewhere in the stacked rows of H generated), then the row is discarded and the process continued. While large n is desired, to ensure a low-density of 1's as described, the parameters t_c , n , and k must be chosen to satisfy

$$\frac{(n - k)(n - k - 1)}{t_c(t_c - 1)} \geq n \quad (11.68)$$

This relation basically forces long n to be used for high-rate codes. On average, codes having such a parity matrix can have very high coding gain. For instance, Richardson and Urbanke have recently noted the average LDPC code parameters (we made adjustments to their deviation from capacity numbers to reflect capacity for an ensemble average of large- n LDPC codes rather than the restricted capacity they studied) in Table 11.37. This table assumes that the codes are used on the AWGN with $\bar{b} < 1$ – that is, transmitted signals are simple ± 1 and there is no issue of shaping gain at such low \bar{b} .

In a paper in IEEE Communication Letters, Chung, Forney, Richardson and Urbanke show that for the binary-input AWGN channel, a rate 1/2 code can be constructed that is within 0.0045 dB of the Shannon capacity. This code did not have a constant number of 1's per row or column (i.e., was not regular), and it appears such non-uniform distribution of 1's is necessary to get extremely close to capacity.

Clearly any specific random construction for an H matrix needs to be explicitly tested to determine its gain and parameters. However, with the results in Table 11.37 to guide design, one could generate codes with confidence that a code with approximately the gains illustrated would be found. Additional coding gain to capacity can be achieved if the inner LDPC system operates at a \bar{P}_b of approximately 10^{-6} or 10^{-7} and an external hard-decoded block code (like a Reed Solomon code of rate .9 or greater) is applied, and will often gain the remaining 1-2 dB of coding gain as well as reduce the probability of error to essentially error-free levels. LDPC codes are based on good distance properties for very complicated codes and do not exhibit the error-flooring effect of turbo codes.

There are various approaches to generation of the H matrix. Gallager originally used a recursive procedure in which smaller LDPC matrices are inserted into larger H matrices. However, this textbook considers codes that appear to be well suited for multilevel transmission, as well as for low-rate transmission, for which the construction follows the random procedure.

Since the error-floor problem associated with interleaving gain in Turbo codes is not evident in good LDPC codes, LDPC codes more uniformly address the issue of increasing minimum distance and simultaneously controlling the growth in nearest neighbors at all small distances. Decoding of LDPC codes follows according to the constraint-decoding methods discussed in Chapter 9.

11.5.1 IBM Array LDPC Codes

IBM researchers E. Eleftheriou, S. Olcer, G. Cherubini, B. Marcus, and M. Blaum recently reported a construction of some LDPC codes that is easy to reproduce and also essentially performs within 1-2 dB of capacity at all \bar{b} . Again, the remaining 1-2 dB can be obtained by serial outer concatenation of a long-length Reed Solomon block code with appropriate longer-length interleaving than the block length of the LDPC code. Shaping gain of 1.53 dB at large \bar{b} (and less at smaller $\bar{b} > 1$) is independent, see Section ??.

The IBM LDPC codes use a Vandermonde matrix structure for H that depends on a $p \times p$ (p is a prime integer) circular shift matrix

$$\alpha = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (11.69)$$

The consequent parity matrix is ALMOST

$$H = \begin{bmatrix} I & I & \dots & I & I \\ I & \alpha & \alpha^2 & \dots & \alpha^{t_r-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ I & \alpha^{\tilde{t}_c-1} & \alpha^{2(\tilde{t}_c-1)} & \dots & \alpha^{(t_r-1)(\tilde{t}_c-1)} \end{bmatrix}. \quad (11.70)$$

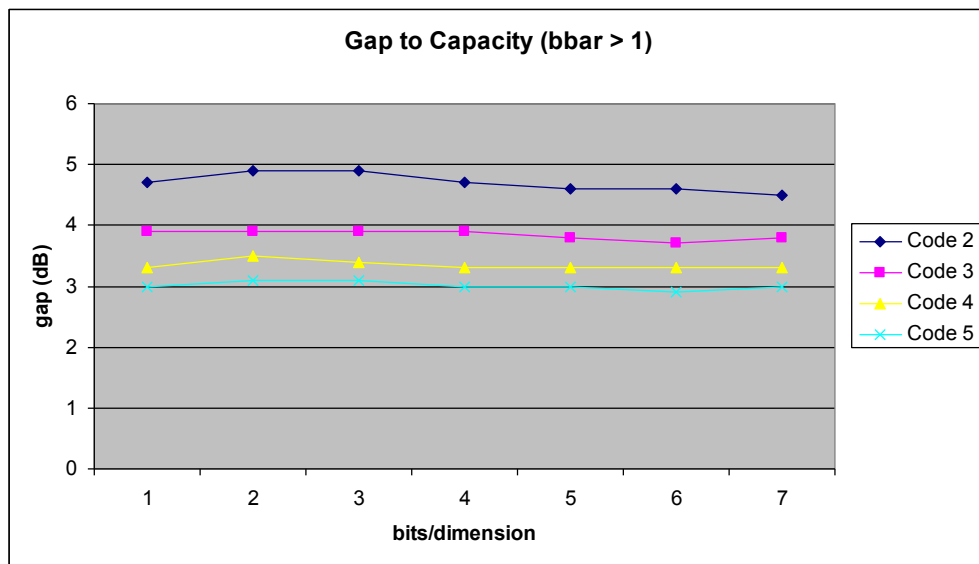
The word ALMOST is capitalized because linearly dependent rows of H (going from top to bottom) are eliminated as they occur, so while the codeword length remains $t_r \cdot p$, the number of rows $n - k < t_c \cdot p$. Table 11.38 lists how many (m) rows of the parity matrix in (11.70) are actually linearly dependent and thus removed. This number is always $m = t_c - 1$. In removing some rows, the number of ones per column reduces in some columns, so that t_c value now represents the maximum number of ones in any column. Some columns have instead $\tilde{t}_c - 1$ ones in them.

The H construction above is known to guarantee aversion of any 4-cycles. The notation \tilde{t}_c is used because the deletion of linearly dependent rows actually reduces t_c in m of the columns and so the codes are not uniform (that is, t_c is not quite constant).

Longer block length typically means higher gain with this construction. Table 11.38 illustrates the codes and parameters when used for $\bar{b} \geq .5$. Both the gap to capacity when the code is used at $\bar{b} < 1$ on the AWGN and when used (as to be shown later in this section) for $\bar{b} > 1$ are enumerated, with the latter being 1.5 dB higher because LDPC codes do not address shaping gain. The gap to capacity varies only slightly with \bar{b} for these codes as illustrated in Figure 11.22.

(n, k)	m	p	t_c	t_r	\bar{b}	Γ at 10^{-7}	high- b Γ	γ_f
(276,207)	2	23	3	12	.7572	3.8 dB	5.3 dB	4.2 dB
(529,462)	2	23	3	23	.8733	3.0 dB	4.5 dB	5.0 dB
(1369,1260)	2	37	3	37	.9204	2.3 dB	3.8 dB	5.7 dB
(2209,2024)	3	47	4	47	.9163	1.8 dB	3.3 dB	6.2 dB
(4489,4158)	4	67	5	67	.9263	1.5 dB	3.0 dB	6.5 dB
(7921,7392)	5	89	6	89	.9332	1.3 dB	2.8 dB	6.9 dB

Table 11.38: IBM Array-Code LDPC parameters.

Figure 11.22: Illustration of nearly constant gap to capacity for integer \bar{b} .

bit label	value	subset number
(0,0,0)	-7	0
(0,0,1)	-5	1
(0,1,1)	-3	3
(0,1,0)	-1	2
(1,0,0)	+1	0
(1,0,1)	+3	1
(1,1,1)	+5	3
(1,1,0)	+7	2

LDPC codes are iteratively decoded using the parity and equality constraint processing of Section 9.6. The IBM researchers noticed that there can be a very large number of processors and this large number is not necessary for constellations with large $\bar{b} > 2$. For such large-constellation applications, the IBM LDPC code is mapped into SQ QAM constellations by directly mapping the encoder output into 1-dimensional constituent components. The modulator takes up to 4 bits at a time from the LDPC encoder output. A useful quantity \bar{b}_o is defined as the number of bits per dimension at the encoder output (so $\bar{b}_o = (n/k) \cdot \bar{b}$). When $\bar{b}_o = .5$, then BPSK is transmitted using each successive LDPC encoder output bit to modulate the polarity, positive or negative (for 1 or 0 respectively). When $\bar{b}_o = 1$, two successive LDPC encoder output bits are mapped into QPSK. When $\bar{b}_o = 1.5$, 3 successive output bits map into 8 SQ QAM, and when $\bar{b}_o = 2$, 4 successive encoder output bits map into 16 QAM. For $\bar{b}_o > 2$ and an integer, only 4 output bits from the LDPC encoder are used: 2 of these bits are used to encode one of 4 subsets in a one-dimensional partitioning, and the other 2 bits are used identically for the other dimension in SQ QAM. The remaining “uncoded” bits are used to select points with the sets - points within a set have an intra-coset distance of nearly 12 dB better than uncoded and thus exceed the gain of the code. An example appears in Table 11.5.1 for the encoding of one of the dimensions of 64 QAM. The first bit or msb is thus not coded by the LDPC encoder. This procedure simplifies decoding (because the additional “uncoded” bits are determined by simple slicing in the receiver), and also reduces the redundancy in the use of the code because there are fewer redundant bits per symbol. Furthermore, the gain is not reduced.

Such a procedure does need to match data rates and redundancies. This development will assume that SQ QAM constellations (or two-dimensional constellations) are used. these constellations can always be viewed as 2 successive PAM constellations as above. The number of bits per 2D symbol is defined as

$$b_2 = \frac{R}{\text{2D symbol rate}} = \frac{q}{p} \quad ; \quad (11.71)$$

The integers q and p may be chosen to approximately arbitrarily closely any data rate and number of bits per QAM symbol. Two situations are of interest: (1) the number of bits is relatively low so that no parallel transitions are used and (2) parallel transitions will be used because the number of bits per QAM symbol is sufficiently large. Figure 11.23 illustrates the first case: In this case, q successive uses of the code produce $n \cdot q$ output bits in response to $k \cdot q$ input bits. This also corresponds to $k \cdot p$ two-dimensional symbols since $kq/b_2 = kp$. A larger number of bits per two-dimensional symbol

$$\tilde{b}_2 = \lceil \frac{n}{k} \cdot b_2 \rceil \quad (11.72)$$

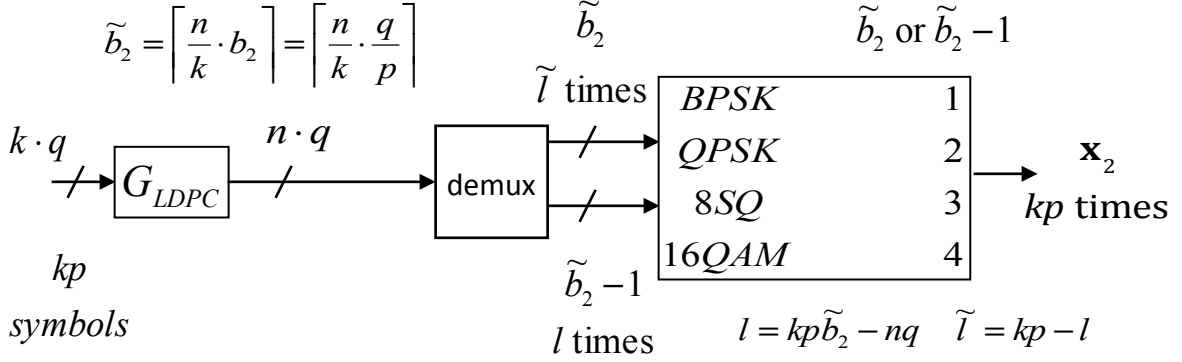
represents the redundancy introduced by the LDPC code. The quantity $\tilde{b}_2 \leq 4$ for the first case, and if $\tilde{b}_2 > 4$, then the second case is considered (as later in this subsection). For the first case, some \tilde{l} 2D symbols will need to carry \tilde{b}_2 bits while l other 2D symbols will carry only $\tilde{b}_2 - 1$ bits. The following equations are solved with \tilde{l} , \tilde{b}_2 , l all integers:

$$\tilde{l} \cdot \tilde{b}_2 + l \cdot (\tilde{b}_2 - 1) = nq \quad (11.73)$$

$$\tilde{l} + l = kp \quad (11.74)$$

An integer solution is

$$l = kp \cdot \tilde{b}_2 - nq \quad (11.75)$$

Figure 11.23: Multi-level use of LDPC code with $\tilde{b}_2 \leq 4$.

$$\tilde{l} = kp - l \quad . \quad (11.76)$$

It is possible that among the q blocks of bits or code uses that up to $q - 1$ individual two-dimensional symbols have bits from two adjacent codewords. The LLR's for the bits of interest are computed as if the other bits are either (1) unknown or (2) the previous decoder's decisions for the earlier block are correct (in which latter case there is a small chance for error propagation).

The second case appears in Figure 11.24. In this case $n \cdot b_2$ input bits are grouped for 4 successive LDPC encoder uses. This number of bits is also equal to $4k + b_{parallel}$ so that, if (n, k) are known/chosen for a particular LDPC (or more generally binary block) code, then

$$b_{parallel} = n \cdot b_2 - 4 \cdot k \quad . \quad (11.77)$$

The symbols are then split between the l of n symbols that carry $b_2 + i$ bits each and the \tilde{l} that carry $b_2 + i + 1$ bits each, where

$$n = l + \tilde{l} \quad (11.78)$$

$$b_{parallel} = l(b_2 - 4 + i) + \tilde{l}(b_2 - 3 + i) \quad . \quad (11.79)$$

Then i is chosen as the smallest integer $i = 4, 5, 6, \dots$ that produces $l \geq 0$ in satisfying

$$l = n \cdot (b_2 - 3 + i) - b_{parallel} \quad . \quad (11.80)$$

Then,

$$\tilde{l} = 4n - l \quad . \quad (11.81)$$

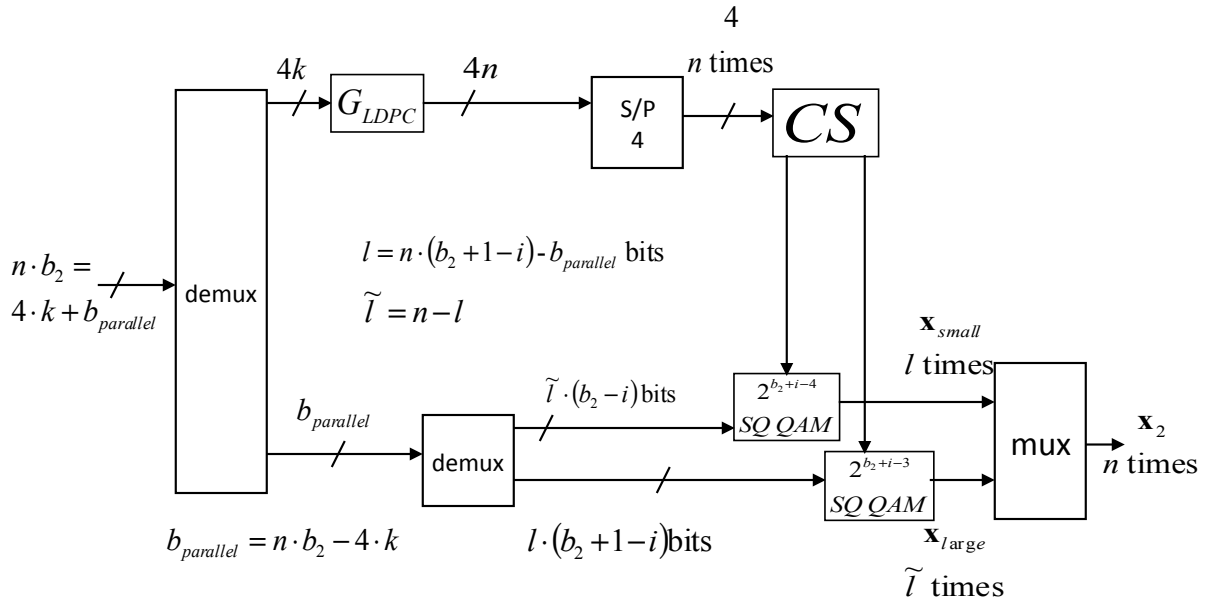
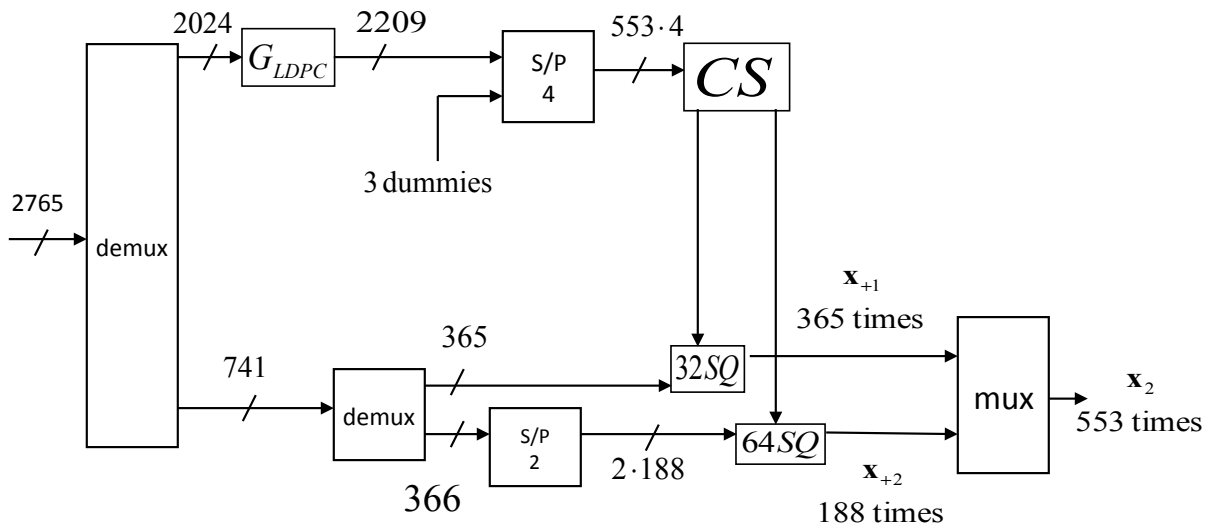
An example occurs in Figure 11.25 for the use of the (2209,2024) IBM code and a data rate of 20 Mbps and a two-dimensional symbol rate of 4 MHz.

Multi-carrier transmission is somewhat easier because it naturally spreads bits over a long symbol interval and many two-dimensional symbols. If B bits per N carriers are used with an (n, k) block code, then

$$\tilde{B} = \lceil B \cdot \frac{n}{k} \rceil \quad (11.82)$$

are loaded. If all carriers have less than 4 bits, then Figure 11.26 illustrates the straightforward implementation. If some carriers have a number of bits exceeding 4, then the loading algorithm is re-run with a FM (or MA) loading algorithm until

$$B' + \sum_{i: b_i > 4} (b_i - 4) = B \quad (11.83)$$

Figure 11.24: Multi-level use of LDPC code with $\tilde{b}_2 > 4$.Figure 11.25: Specific 20 Mbps example of LDPC code with $b_2 = 5$.

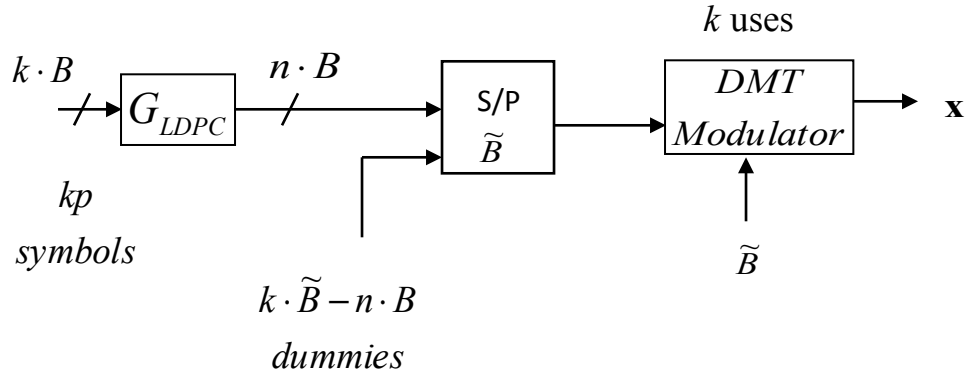


Figure 11.26: Use of LDPC with multi-carrier loading.

where B' solves

$$\lceil \frac{n}{k} \cdot B \rceil = 4 \cdot N_{>4} \sum_{i: b_i \leq 4} b_i \quad (11.84)$$

The number of parallel bits is

$$b_{parallel} = \sum_{i: b_i > 4} (b_i - 4) \quad (11.85)$$

11.5.2 Feeding the Decoder

To obtain the initial intrinsic probability from an AWGN received channel output value, the mapping allows each bit's value of the encoder output to be identified with one or more points in the constellation. This set of points is called either $X_{v_m=1}$ or $X_{v_m=0}$. The probability is obtained by the following sum

$$p(v_k = 1) = \sum_{x \in X_{v_k=1}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-x)^2} \cdot p_x \quad (11.86)$$

A similar sum is computed for $v_k = 0$. Then, LDPC decoding is initialized with the prior given by

$$LLR_k = \ln \frac{p(v_k = 1)}{p(v_k = 0)} \quad (11.87)$$

11.5.3 Lee's LDPC Routines

2006 EE379B student Chien-Hsin Lee has graciously provide the software routines to construct various LDPC codes, both parity and systematic generator matrices as well as an encoder and decoder function. The matlab software is at the web site:

To generate the LDPC H matrix for any IBM code, the function `get_h_matrix` is useful. This function called by `H = get_h_matrix(prime,tr,tc,m)` has 4 inputs

1. prime = the prime number
2. tr = the number of ones per row
3. tc = the maximum number of ones per column
4. m = the number of dependent rows removed (from Tables earlier $m = t_c - 1$)

The sole output is the parity matrix.

A second routine provides the systematic encoder and is `enc [H G] = systematic(H)`. The inputs and outputs are obvious.

A third routine provides the encoder output and also the AWGN channel outputs when noise of a specified SNR is added: [rx_bit, coded_bit, message_bit] = encoder(SNR_dB,G,random,message_bit). The inputs and outputs are again obvious.

And, finally a decoder routine specified by [decoded_bits,llr_bits, iter] = ldpc_decoder_spa(H,bits,max_iter,var,fast). var is the variance of the noise, and fast=1 stops the decoder if the parity check equations are all satisfied, while fast=0 means execute max_iter iterations of decoding.

The program listings appear to end this subsection:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function [H_no_dep H] = get_h_matrix(p,rw,cw,first_1_start);
%% Generate LDPC H Matrix Uses IBM's Method As Per Cioffi's Class Note
%% Example: to Generate (529,462) code, p=23, rw=23, cw=3, first_1=2
%%           H = get_h_matrix(23,23,3,2),
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of input variables
%% p      : Prime number of the size of base matrix of size p-by-p
%% rw     : Row weight equals to number of base matrix per row, eq to K
%% cw     : Column weight equals to number of base matrix per column,eq to J
%% first_1: Set to 2 per IBM's Method right shift 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of output variables
%% H_no_dep : the parity check matrix with no dependent rows
%% H        : without removing the dependent rows
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% To Do :
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% EE379B, Chien-Hsin Lee, 06/2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [H_no_dep] = get_h_matrix(p,rw,cw,first_1_start);
% generate base matrix
a = eye(p,p);
a = [a(first_1_start:p,:);a(1:first_1_start-1,:)];

% generate H matrix
H = [];
for row = 1:cw
    current_row = [];
    for cl = 1:rw
        current_row = [current_row,a^((row-1)*(cl-1))];
    end
    H = [H;current_row];
end

% remove dependent rows
% this need to be update with mod2 operation
[Q R] = qr(H);
H_no_dep = [];
for i = 1:p*cw
    if( abs(R(i,i)) > 1E-9)
        H_no_dep = [H_no_dep;H(i,:)];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function [H_sys, G_sys] = get_g_matrix(H);
```

```

%% This routine remove the dependent row from original H matrix.
%% Find the systematic G matrix and the H matrix work with this G matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of input variables
%% H      : parity check matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of output variables
%% H      : H to work with systematic G matrix
%% G      : Systematic G matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% To Do :
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% EE379B, 06/2006, Chien-Hsin Lee
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [H_final, G_sys] = systematic(H);
[ row col ] = size(H);
H_in      = H;
offset    = 0;
r = 1; c = 1;
num_dep_row = 0;
%% Gaussian Elimination PART1
%% This find the leading 1 in each column
%% and eliminate the 1s in the row below
for i = 1:row
    row_order(i) = i;
end
while( r <= row && c <= col)
    row_lst = find(H(r:row,c));
    if( isempty(row_lst))
        c = c + 1;
    else
        if( length(row_lst) > 1)
            for i = 2:length(row_lst)
                H(row_lst(i)+r-1,:) = mod(H(row_lst(1)+r-1,:) + H(row_lst(i)+r-1,:),2);
            end
        end
        H_temp      = H;
        H_temp(r,:) = H(row_lst(1)+r-1,:);
        H_temp(row_lst(1)+r-1,:) = H(r,:);
        H           = H_temp;
        row_temp    = row_order;
        row_temp(r) = row_order(row_lst(1)+r-1);
        row_temp(row_lst(1)+r-1) = row_order(r);
        row_order   = row_temp;
        r = r + 1;
        c = c + 1;
    end
end
r = row;
while( H(r,:) == 0 & r ~= 0)
    H(r,:) = [];
    r = r-1;
end
num_dep_row = row-r;

```

```

%flg = sprintf('Report: %d dependent rows are removed;\n Old matrix is %d by %d;\n New matrix is %
%disp(flq);
row = r;

%% Column Permutation To Put Diagonal In All 1's
for c = 1:col
    column(c) = c;
end
r = 1; c = 1;
while( r <= row && c <= col)
    if( H(r,c) ~= 1)
        H(:,c:col) = [H(:,c+1:col),H(:,c)];
        column(c:col) = [column(c+1:col),column(c)];
        r = r-1; c = c-1;
    end
    r = r+1; c=c+1;
end

%% Gaussian Elimination Part II To Get The Systematic H
for i = 0:row-1
    %% find backward leading cols and eliminate the upper triangle 1s
    col_lst = find(H(:,row-i));
    if( length(col_lst) > 1)
        for j = 1:length(col_lst)-1
            H(col_lst(j),:)=mod(H(col_lst(j),:)+H(row-i,:),2);
        end
    end
end

%% Get G And H Matrix
G_sys = [H(:,row+1:col)',eye(col-row)];
H_no_dep = [];
if(num_dep_row > 0)
    x = row_order(row+1:row+num_dep_row);
    x = sort(x);
    j = 1;
    for i = 1:row+num_dep_row
        if(i~=x(j))
            H_no_dep = [H_no_dep;H_in(i,:)];
        else
            j=j+1;
        end
    end
else
    H_no_dep = H_in;
end

for i = 1:col
    H_final(:,i)=H_no_dep(:,column(i));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function [encoded_bit message_bit] = encoder(SNR,G,random,message_bit)
%% LDPC Encoder

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of input variables
%% SNR_dB      : Receiver SNR in dB
%% G           : Generator matrix
%% random      : encoder generate random bit streams
%% message_bit : message_bit to be encoded.
%%            : if random is 1, message_bit is ignored
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of output variables
%% rx_bit      : coded_bit + AWGN
%% coded_bit    : encoded bit
%% message_bit : message bit to be encoded
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% To Do :
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% EE379B, 06/2006, Chien-Hsin Lee
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [rx_bit, coded_bit, message_bit] = encoder(SNR_dB,G,random,message_bit)

[k,n]      = size(G);
if(random == 1)
    message_bit = round(rand(1,k));
end

noise      = randn(1,n)/sqrt(10^(SNR_dB/10));
coded_bit  = mod(message_bit*G,2);
modulated_bit= coded_bit*2-1;
rx_bit     = modulated_bit + noise; %% sending -1 only

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function [decoded_bits llr_bits iter]=ldpc_decoder_spa(H,bits,max_iter,var,fast)
%% SPA LDPC Decoder
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of input variables
%% H           : parity check matrix
%% bits        : Received bits
%% max_iter    : maximum number of iteration
%% var         : variace of the noise
%% fast        : 1: stop at good parity, 0: run till maximum iteration.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definition of output variables
%% decoded_bit : decoded bits
%% llr_bit     : llr of each input bits
%% iter        : number of iteration used
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% To Do :
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% EE379B, 06/2006, Chien-Hsin Lee
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [decoded_bits,llr_bits, iter ] = ldpc_decoder_spa(H,bits,max_iter,var,fast)
[ row col ] = size(H);
iter       = 0;
pass       = 0;

```



```

llr_int    = zeros(1,col);    % intrinsic
llr_b2c    = zeros(row,col); % bit node to check node
llr_c2b    = zeros(row,col); % check node to bit node
tr         = (-1).^mod(sum(H')',2);

%% intrinsic bit probability
for i = 1:col
    llr_int(i) = ((bits(i)+1)^2 - (bits(i)-1)^2)/2/var;
end
llr_bits    = llr_int;
decoded_bits(i) = (sign(llr_bits(i))+1)/2;

while(iter < max_iter && (pass ~= 1 || fast == 0) )
    iter = iter + 1;
    %% update bit note to check node LLR
    for j = 1:col
        row_ptr = find(H(:,j));
        for i = 1:length(row_ptr)
            llr_temp = llr_int(j);
            for k = 1:length(row_ptr)
                if( k ~= i)
                    llr_temp = llr_temp + llr_c2b(row_ptr(k),j);
                end
            end
            llr_b2c(row_ptr(i),j) = llr_temp;
        end
    end
end

%% update probabiltly of check equation
for i = 1:row
    col_ptr = find(H(i,:));
    for j = 1:length(col_ptr);
        X = 1;
        for k = 1:length(col_ptr);
            if( k ~= j)
                X = X*tanh(llr_b2c(i,col_ptr(k))/2);
            end
        end
        llr_c2b(i,col_ptr(j)) = tr(i)*2*atanh(X);
    end
end

%% check result
for i=1:col
    llr_bits(i)    = llr_int(i)+sum(llr_c2b(:,i));
    decoded_bits(i) = (sign(llr_bits(i))+1)/2;
end
pass    = 1-sum(mod(decoded_bits*H',2));
end %% while()

```

Exercises - Chapter 11**11.1 Interleaver**

Assume a generalized triangular interleaver throughout this problem:

- (3 pts) Find $G(D)$ and $G^{-1}(D)$ for a generalized triangular interleaver with $J = 4$ and $K = 7$.
- (1 pt) compute delay (end to end)
- (1 pt) How much does the gain (for bursty error) of hard convolutional code appear multiplied if error bursts are infrequent.

11.2 Serial Code Concatenation

Assume two convolutional codes are concatenated serially with rates \bar{b}_1 for the inner code and \bar{b}_2 for the outer code.

- (1 pts) Find the overall \bar{b} for the concatenated system.
- (1 pt) Suppose the inner code is a trellis code and repeat part a.
- (2 pts) How does interleaving affect the rates of the two systems in parts a and b?
- (1 pt) Suppose the parity bits added by a systematic encoder in part a are not re-encoded by the outer code - then what is the rate of the concatenated system?

11.3 Convolutional Interleaving - Final 2003 (24 pts)

A generalized triangular interleaver designed for bytes (symbols) has period $L = 7$ and depth $J = 5$.

- Find the Generator Matrix for the interleaver. (2 pts)
- Find the inverse generator matrix for the de-interleaver. (2 pts)
- Draw the interleaver and de-interleaver and show use of byte-wide registers as D and provide the clock rate at which these registers pass information. A cascade of m D registers can be abbreviated D^m . How many bytes of memory are used? (2 pts)
- What is the delay through the interleaver and de-interleaver. (1 pt)
- Assuming the byte-level code associated with this interleaver corrects d_{free} bytes in error - what is the number of bytes that can be corrected from burst disturbances with the interleaver (if the burst is 35 samples or less). (1 pt)

11.4 Interleaving and Wireless - 10 pts

A wireless channel uses a 16 QAM modulator to transmit data. The symbol clock is 4 MHz. In addition to the channel being an AWGN, there is a gain on each channel that can vary with symbol period. When in a nominal state, the channel has SNR=18 dB. In a fading state, the SNR is either 21 dB with probability 0.999 or -9 dB with probability .001. Each of the gains is independent of the others.

- What is the data rate? (1 pt)
- What is the probability of symbol error in the nominal state? (1 pt)
- What is the probability of a symbol error in the fading state? (2 pts)
- Suppose that a 50 and that the channel is always in the fading state. Use the best 4-state $d = 5$ convolutional code as an outer code with hard decoding and design an interleaving scheme of minimal end-to-end delay for this code that ensures that the probability of symbol error that is less than 10^{-6} . (6 pts)

11.5 Zero Stuffing and block iterative decoding

Two convolutional codes are concatenated in parallel with rates \bar{b}_1 for the inner code and \bar{b}_2 for the outer code. The first code has constraint length ν_1 and the second code has ν_2 . A few bits are placed at the end of a length- L packet to force the first encoder's state to zero. For parts c through f, assume $\bar{b}_1 = 1/n_1$ and $\bar{b}_2 = 1/n_2$, and that the two codes are systematic.

- (2 pts) How many bits are wasted to force the first encoder's state to zero? When is this number insignificant?
- (2 pts) Do these bits also force the second code to state zero in general? What might we do to also force this code to state zero?
- (2 pts) Suppose instead of separately computing the likelihood or probability densities for input zero and one bits independently, could we instead propagate the log of the likelihood ratio? (the likelihood ratio is the ratio of the probability density of a "1" to a "0"). What would be the resultant eventual detection rule using some set of Likelihood ratios at some stage in decoding?
- (3 pts) Find an iterative expression for the alternative of propagating the log likelihood ratio of part c in iterative decoding, instead of propagating the Likelihood function itself. Explicitly enumerate the values of the intrinsic log likelihood ratio for the assumption of a uniform input and transmission over a BSC with parameter p .
- (5 pts) Find an iterative expression for the alternative of propagating the log likelihood ratio of part c in iterative decoding, instead of propagating the Likelihood function itself. Explicitly enumerate the values of the intrinsic log likelihood ratio for the assumption of a uniform input and transmission over an AWGN with noise σ^2 and bipolar transmission levels ± 1 on the code outputs.
- (1 pt) Comment on propagation of the likelihood ratio as an alternative to propagation the distributions themselves.

11.6 Schedules – 11 pts

A given triangular interleaver has depth $J = 3$ and period $L = 4$.

- What is the minimum number of memory cells? (1 pt)
- Create a scheduling diagram similar to those in Tables 11.3 and 11.4 for this triangular interleaver? (5 pts)
- How many different length schedules are there in your answer for part a? (1 pt)
- Compute S and compare to the sum of the lengths of the different-length schedules. (1 pt)
- Explain the condition in Section 11.2 that $\frac{mL}{J-1}$ should not be an integer if the number of periods is smaller than half the delay minus one. (3 pts)

11.7 Puncturing and Generators – 10 pts

The same rate $2/3$ convolutional code is used twice in a parallel concatenation system.

- What is code rate of the concatenated system with no puncturing? (1 pts)
- Find a puncturing scheme to make each of the convolutional codes rate $3/4$ and what is the new rate of the concatenated system? (3 pts)
- What is the size of the generator matrix that describes the puncturing? (1 pt)
- Show the G matrix for your scheme. (3 pts)
- Repeat part b for a new convolutional code rate of $4/5$? (2 pts)

11.8 Puncturing and Generators – 10 pts

The same rate $2/3$ code is used twice in a serial concatenation system.

- What is code rate of the concatenated system with no puncturing? (1 pts)
- Find a puncturing scheme to make each of the convolutional codes rate $3/4$ and what is the new rate of the concatenated system? (3 pts)
- What is the size of the generator matrix that describes the puncturing? (1 pt)
- Show the G matrix for your scheme. (3 pts)
- Repeat part b for a new convolutional code rate of $4/5$? (2 pts)

11.9 Turbo-Coded Trits (Final 2001): (12 pts)

Prof. Bo Zhao of the University of California at Bear has again escaped trans-bay security and is loose on Sand Hill Road in Menlo Park. He has invented 3-level logic for integration on circuits, thus allowing the use of trits, which measure the data content of a constellation according to

$$t = \log_3(M) \quad . \quad (11.88)$$

The levels that can be processed are 0,1, and 2, which translated in logic into voltages -1, 0, and +1. An earlier success with 3-way trellis code partitioning has encouraged him to investigate turbo codes and what he calls LDTC (low density trinity codes). He starts with the design of a linear convolutional code based on 1 trit in and two trits out.

- Compute the code rate \bar{t} in trits per dimension. (1 pt)
- Using up to 9 states, design a trellis that provides maximum distance between codeword sequences? (3 pts)
- Find the coding gain in a manner analogous to binary convolutional codes. (1 pt)
- Find a systematic $G(D)$ and corresponding “trinity” matrix for your code in part b). (2 pts)
- The same code with systematic realization is used in two parallel-concatenated codes with a uniform random interleaver between them. Find an approximate expression for the probability of bit error assuming iterative decoding with the APP algorithm on each decoder. (Hint, you may have to investigate the error event corresponding to d_{\min} and use some approximation to determine how likely it is to affect a second decoder) (2 pts)
- (Zhao also claims it is possible to use 3-level logic for the trinity matrix of any ternary linear code. Find relations for the extrinsic probability of an equality constraint and for a “trinity” constraint. Can you say something about the use of LLR here and whether it makes sense? (2 pts)
- Do you see any advantage for the 3-level codes with respect to binary codes when used in the turbo or LDTC context? What would you do to Zhao if you were a venture capitalist working on Sand Hill Road? (1 pt)

11.10 High-performance Turbo Coding - 9 pts - Final 2003

An AWGN channel uses QAM with symbol rate fixed at 1 MHz and 21 dB of SNR.

- What is the highest data rate that can be achieved with the 16-state 4D Wei code (ok to use gap approximation here) at $\bar{P}_e = 10^{-6}$? (1 pt)
- Using the AT&T Turbo code, repeat part a. (1 pt)
- For the AT&T Turbo code, show the encoder with an PRBS-based approximation to a uniform random interleaver, constellation with bit labels.

- d. For the receiver using the APP-based iterative-decoding (you may assume a very long block length so that essentially one block is decoded and complexity per symbol time is constant):(4 pts)
- (i) How many states are in the trellises for each of the APP decoders used? (1 pt)
 - (ii) Assuming one multiply-and-add is an operation, how many operations are used in each of the APP decodings per symbol? (1 pt)
 - (iii) If iterative decoding converges after 5 cycles (of both decoders), what is the total number of operations per second for the iterative decoding? (1 pt)
 - (iv) How many operations are necessary to compute input prior probabilities from the constellation (you can assume table look of a probability density evaluation or of a log function is one operation)? (1 pt)
 - (v) Using your answer in part d, compare the complexity in operations per second to the complexity in operations per second for the 4D 16-state Wei code. What might be done to reduce the complexity without compromising performance too much? (2 pts)

11.11 SBS detection with and without LDPC coding on EPR4 channel: (20 pts)

In this problem you are asked to use matlab to compare uncoded SBS detection and SBS detection with LDPC coding at the output of an EPR4 channel with AWGN. The matlab files that you need for this problem are available on the class web page.

Assume the input is binary and is equally likely. A single frame of the input sequence of length 1088 bits that you need to use for this exercise is given in the file `m.mat`. We would like to pass this frame through the channel `EPR4channel.m` 16 times and find the average BER. The function `EPR4channel.m` takes d , the distance between constellation points, and the frame number i ($i = 1, 2, \dots, 16$) as arguments, in addition to input frame $\mathbf{xFrame}_i(D)$, and returns output frame $\mathbf{yFrame}_i(D)$. Assume $\mathcal{E}_b/\mathcal{N}_0 = 5$ dB, unless mentioned otherwise. Adjust \mathcal{E}_x appropriately to get the required value of $\mathcal{E}_b/\mathcal{N}_0$.

- a. (2 pts) What is the number of training symbols required to estimate the variance of the AWG noise of the channel to within 0.05 dB and with 90% confidence?
- b. (2 pts) Use a training sequence of the length found in part a to estimate noise variance of the channel `EPR4channel.m`.
- c. (1 pt) Use precoding and 2-level PAM modulation. After going through the channel, use uncoded SBS detection to detect the information bits. Find the number of bit errors in the detected sequence over the 16 transmitted frames.
- d. (1 pt) Now let us add LDPC coding. Use the LDPC code given in the file `ldpc_code.mat`, which has a 136×1224 parity check matrix that has $t_c = 3$ ones in each column. Pass the input data frame through `encoder.m` to get the LDPC codeword of length $N = 1224$. This codeword will be transmitted through the channel 16 times for BER calculations. (1 pt)
- e. (1 pt) Pass the codeword through a binary precoder and 2-level PAM modulator, and then go through the `EPR4channel.m`.
- f. (1 pt) For each received symbol y_k , $k = 1, \dots, N$, where N is the codeword length, calculate the channel probability densities p_{y_k/\tilde{y}_k} for each possible value of \tilde{y}_k , where y_k is the noiseless channel output. Normalize these conditional probabilities so that they add up to 1.
- g. (2 pts) Calculate the a priori probabilities $p_k(0)$ and $p_k(1)$, $k = 1, \dots, N$, by adding the corresponding channel probabilities found in part (f) based on the mapping of SBS detection.
- h. (1 pt) Calculate the log likelihood ratio LLR_k , $k = 1, \dots, N$. To avoid dividing by zero use: $\text{LLR}_k = \log \left(\frac{\max(p_k(1), \epsilon)}{\max(p_k(0), \epsilon)} \right)$, where $\epsilon = 10^{-5}$.

- i. (4 pts) Use the function `ldpc_iteration`, which performs a single LDPC decoding iteration, to decode the received codeword. Initialize the messages from checks to bits to all zeros, and use the a priori LLR sequence found in part h. The vector `message_bit_locations` in `ldpc_code.mat` indicates the locations of message bits in the LDPC codeword. Find the number of bit errors in the detected information sequence over the 16 transmitted frames, for the cases of 1, 3, and 10 LDPC decoding iterations.
- j. (3 pts) Plot the BER versus $\mathcal{E}_b/\mathcal{N}_0$ for uncoded SBS detection and LDPC coded SBS detection with 1, 3, and 10 decoding iterations over the $\mathcal{E}_b/\mathcal{N}_0$ range 3 to 8 dB. Use a step of 0.2 dB in your final plots.
- k. (2 pts) What is the LDPC coding gain with 1, 3, and 10 decoding iterations at $\text{BER}=10^{-3}$ relative to uncoded SBS detection.

11.12 Multilevel LDPC decoding on AWGN channel (22 pts)

This problem uses matlab to compare the performance of uncoded transmission and symbol-by-symbol detection with the performance of an LDPC code and iterative decoding at the output of the same AWGN channel. The information rate is with 8-level PAM transmission. The matlab files that you need for this problem are available on the class web page. The uncoded 8-level PAM constellation uses the following “Gray-code” bit mapping:

m_k, m_{k+1}, m_{k+2}	000	001	011	010	110	111	101	100
constellation point:	$-7d/2$	$-5d/2$	$-3d/2$	$-d/2$	$d/2$	$3d/2$	$5d/2$	$7d/2$

Each binary in-

put message is equally likely. A single frame of the binary input sequence of length 1089 bits for this exercise is given in the file `m.mat`. This exercise passes this frame through the channel `AWGNchannel.m` 16 times and finds the corresponding average BER. The function `AWGNchannel.m` takes the frame number i ($i=1, 2, \dots, 16$) as an argument, in addition to input frame `xFrame.i(D)`, and returns output frame `yFrame.i(D)`. Assume $\frac{\mathcal{E}_b}{\mathcal{N}_0} = \frac{\text{SNR}}{2b} = 16\text{dB}$, unless mentioned otherwise and adjust \mathcal{E}_b appropriately to get the required value of $\frac{\mathcal{E}_b}{\mathcal{N}_0}$.

- a. What is the number of training symbols required to estimate the variance of the AWG noise of the channel to within 0.075dB of its true value with 98% confidence? (2 pts)
- b. Use a training sequence of the length found in part a to estimate noise variance of the channel `AWGNchannel.m`. (2 pts)
- c. Use uncoded 8-level PAM modulation given above. Use SBS detection to detect the information bits from the output of the AWGN. Find the number of bit errors in the detected sequence over the 16 transmitted frames. (2 pts)
- d. Use uncoded 8-level PAM modulation given above. Use SBS detection to detect the information bits from the output of the AWGN.
- e. Find the number of bit errors in the detected sequence over the 16 transmitted frames. (2 pts)
- f. Pass the codeword through the 8-level PAM modulator, and then go through the `AWGNchannel.m`. Calculate the loss in dB caused by the any bandwidth expansion that would be necessary to accommodate the rate loss in part d. (1 pt)
- g. For each received symbol y_k , $k = 1, \dots, N/3$, where N is the codeword length, calculate the channel probabilities p_{y_k/\tilde{y}_k} for each possible value of \tilde{y}_k , where \tilde{y}_k is the noiseless channel output. Normalize these conditional probabilities so that they add up to 1. (3 pts)
- h. Calculate the intrinsic probabilities $p_k(0), p_k(1), p_{k+1}(0), p_{k+1}(1), p_{k+2}(0), p_{k+2}(1), k = 1, \dots, N/3$, by adding the corresponding channel probabilities found in part f based on the symbol to bit demapping shown above. (3 pts)

- i. Calculate the log likelihood ratio LLR_k , $k = 1, \dots, N$. To avoid dividing by zero use:

$$\text{LLR}(k) = \log \left(\frac{\max(p_k(1), \epsilon)}{\max(p_k(0), \epsilon)} \right), \quad (11.89)$$

where $\epsilon = 10^{-5}$. (1 pt)

- j. Use the function `ldpc_iteration`, which performs a single LDPC decoding iteration, to decode the received codeword. Initialize the LLR messages/extrinsic-probabilities from parity check nodes to equality nodes to all be zero, and use the a priori LLR sequence found in part h to initialize the decoding process. The vector `message_bit_locations` in `ldpc_code.mat` indicate the locations of message bits in the LDPC codeword. Find the number of bit errors in the detected information sequence over the 16 transmitted frames, for the cases of 1, 3, and 10 LDPC decoding iterations. (2 pts)
- k. Plot the BER versus $\frac{\mathcal{E}_b}{N_0}$ for uncoded SBS detection and LDPC coded SBS detection with 1, 3, and 10 decoding iterations over the $\frac{\mathcal{E}_b}{N_0}$ range 14 to 21 dB. Use a step of 0.2dB in your final plots. Comment on the slight loss in part e and how it affects the result here. (3 pts)

11.13 Combined APP and LDPC decoding on EPR4 channel - Al-Rawi's Triathlon: (24 pts)

This problem uses matlab to compare uncoded SBS detection and combined APP soft detection with LDPC soft decoding at the output of an EPR4 channel with 2-level PAM transmission. The matlab files that are necessary for this problem are available on the class web page.

In this iterative scheme, extrinsic soft information is passed back and forth between the APP soft-channel detector and the soft LDPC decoder. Number of iterations is expressed as $(N_{channel}, N_{ldpc})$, where $N_{channel}$ is the number of passes through the APP detector and LDPC decoder. Each pass through the LDPC decoder uses N_{ldpc} decoding iterations. Extrinsic output of the APP soft-channel detector is used as a priori input to the LDPC decoder, and if $N_{channel} > 1$, the extrinsic output of the LDPC decoder is used as a priori input to the APP detector. The final output is always taken from the LDPC decoder. Notice that the APP detector uses raw probabilities and the LDPC decoder uses probabilities in LLR domain, so you need to do the necessary conversions when passing information back and forth between the two. When converting from LLR to probability, you need to truncate your LLR values so as to keep them within the range from 100 to +100, to avoid getting infinite values in matlab.

Assume the binary input is equally likely. A single frame of the binary input sequence of length 1088 bits that you need to use for this exercise is given in the file `m.mat`. We would like to pass this frame through the channel `EPR4channel.m` 16 times and find the average BER. The function `EPR4channel.m` takes `d`, the distance between constellation points, and the frame number `i` (`i=1, 2, 16`) as arguments, in addition to input frame `xFrame.i(D)`, and returns output frame `yFrame.i(D)`. Assume $\frac{\mathcal{E}_b}{N_0} = 0.4\text{dB}$, unless mentioned otherwise. Adjust \mathcal{E}_b appropriately to get the required value of $\frac{\mathcal{E}_b}{N_0}$.

- What is the number of training symbols required to estimate the variance of the AWG noise of the channel to within 0.15dB and with 99.9% confidence? (2 pts).
- Use a training sequence of the length found in part a to estimate noise variance of the channel `EPR4channel.m`. (2 pts)
- Use precoding and 2-level PAM modulation. After going through the channel, use uncoded SBS detection to detect the information bits. Find the number of bit errors in the detected sequence over the 16 transmitted frames. (1pt)
- Plot the BER versus $\frac{\mathcal{E}_b}{N_0}$ for uncoded SBS detection over the \mathcal{E}_b range 0 to 8 dB. Use a step of 0.2dB. (1 pt)
- Use the LDPC code given in the file `ldpc_code.mat`, which has a 136×1224 parity check matrix that has $t_c = 3$ ones in each column. Pass the input data frame through `encoder.m` to get the LDPC codeword of length $N = 1224$. This codeword will be transmitted through the channel 16 times for BER calculations. (1 pt)

- f. Pass the codeword through a binary precoder and 2-level PAM modulator, and then go through the `EPR4channel.m`. What is the rate loss? How many additional bits would need to be augmented to the end of the channel input sequence to force a return to state zero in the EPR4 trellis? Are these bits necessary here? (1 pt)
- g. For each received symbol y_k , $k = 1, \dots, N$, where N is the codeword length, calculate the channel probabilities p_{y_k/\tilde{y}_k} for each possible value of \tilde{y}_k , where \tilde{y}_k is the noiseless channel output. Normalize these conditional probabilities so that they sum to 1. (1 pt)
- h. Use the APP algorithm for soft detection, followed by LDPC soft decoding in the iterative scheme explained above. Assume that each APP detection iteration requires $22N \cdot 2^\nu$ computational operations, and each LDPC decoding iteration requires $(8t_c - 1)N$ computational operations, where N is the codeword length, and 2^ν is the number of states in the trellis. What is the minimum complexity to get zero bit errors in the 16 received frames. (5 pts)
- i. Assuming we are transmitting at 1Mb/s rate, express this complexity of part h in number of operations per second (ops/sec). (2 pts)
- j. Find the number of bit errors in the detected sequence over the 16 frames for the following iterative schemes $(N_{channel}, N_{ldpc}) = (1, 1), (1, 3),$ and $(3, 1)$. (3 pts)
- k. Plot the BER versus $\frac{\mathcal{E}_b}{N_0}$ for combined soft APP detection and LDPC decoding with the schemes in part j over the \mathcal{E}_b range 0 to 2 dB. Use a step of 0.1dB in your final plots. (3 pts)
- l. What is the coding gain of iterative schemes in part j over SBS detection at $\bar{P}_b = 10^{-3}$. (2 pts)

11.14 *Are you a Jedi Knight yet? - Final 2003 - 12 pts*

You are a young Stanford graduate in the year 2134 and have just joined the Alliance for Intergalactic Liberty as a transmission engineer. An evil empire is jamming transmissions that are crucial to the Alliance survival, leaving an AWGN channel with SNR=15 dB for wireless transmission with a fixed symbol rate of $1/T = 10$ MHz. Your Alliance friends need the highest possible data rate you can provide them on this link.

- a. What is the maximum data rate with the fixed symbol rate? (1 pts)
- b. Design code and transmission system that achieves 90% of this highest rate at low probability of bit error. (10 pts - better designs will earn more points)
 - (i) show or describe your constellation
 - (ii) describe or illustrate your encoder(s) and any significant parts
 - (iii) describe the receiver
- c. Suppose a very large look-up table is allowed in both the transmitter and the receiver, and a delay of 100 milliseconds is not of concern and you can somehow then get 1 dB of additional shaping gain. How much higher might you transmit? (1 pt)

11.15 *Puncturing and Generators - 10 pts*

An AWGN has an SNR of 14.2 dB at some given symbol rate. The probability of error goal for QAM transmission with $b = 4$ is $\bar{P}_b \leq 10^{-6}$. (For parts b and c, use Turbo Codes, for parts d,e, and f, use LDPC codes, and for part g use either.)

- a. What is the capacity c in bits/symbol of this channel? (1 pt)
- b. Design a turbo code at this same symbol rate that meets the data rate and probability of error goal. Show the encoders, interleavers, and constellation. (5 pts)
- c. For the code in part b, what is a reasonable size for the interleaver period? What is a good value for S if this interleaver is an S-random interleaver? (2 pts)

- d. Repeat part b using an IBM LDPC code and show exact numbers of bits, dummy bits, etc in the encoder. (6 pts)
- e. How many parity checks are there in your code and how many bits contribute to each? (2 pts)
- f. In the decoder, how many equality nodes are used? (2 pts)
- g. What could you do to double the data rate achieved in parts b or d at the same probability of error? (2 pts)

Bibliography

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

TQ DELTA, LLC, v. 2WIRE, INC.	Plaintiff, Defendant.	Civil Action No. 13-cv-1835-RGA
TQ DELTA, LLC, v. ZHONE TECHNOLOGIES, INC.	Plaintiff, Defendant.	Civil Action No. 13-cv-1836-RGA
TQ DELTA, LLC, v. ZYXEL COMMUNICATIONS, INC. and ZYXEL CORPORATION,	Plaintiff, Defendants.	Civil Action No. 13-cv-2013-RGA
TQ DELTA, LLC, v. ADTRAN, INC.	Plaintiff, Defendant.	Civil Action No. 14-cv-954-RGA
ADTRAN, INC., v. TQ DELTA, LLC.	Plaintiff, Defendant.	Civil Action No. 15-cv-121-RGA

**DECLARATION OF DR. CHRISSAN IN SUPPORT OF PLAINTIFF'S OPENING
CLAIM CONSTRUCTION BRIEF FOR FAMILY 6 PATENTS**

I, Douglas A. Chrissan, hereby declare under penalty of perjury:

1. I have been retained by TQ Delta, LLC (“TQD”) as an expert in connection with the above-captioned litigation to offer opinions regarding certain claim construction issues for U.S. Patent Nos. 8,462,835 (“the ’835 patent”) and 8,594,162 (“the ’162 patent”), hereinafter “the Family 6 patents.”

2. I submit this declaration in support of TQD’s Opening Claim Construction Brief. I have been asked to provide my opinions as to how one of ordinary skill in the art would understand certain claim terms appearing in the Family 6 patents.

3. I am being compensated as an independent consultant in this matter at the rate of \$250 per hour for analysis of documents and preparation of any declaration or report. This compensation is not dependent on my opinions or testimony, or the outcome of this litigation.

I. QUALIFICATIONS AND BACKGROUND

4. I am presently a technical consultant in the areas of communications systems, multimedia systems, computer systems and digital signal processing.

5. I earned a B.S. and M.S. in Electrical Engineering from the University of Southern California in 1988 and 1990, respectively, and a Ph.D. in Electrical Engineering from Stanford University in 1998.

6. A copy of my *curriculum vitae* is included with this declaration.

7. I was a Masters Fellow and Member of the Technical Staff at Hughes Aircraft Company in El Segundo, California, from 1988–1993. While at Hughes Aircraft, I designed and developed communication systems for commercial and military spacecraft, including for the MILSTAR satellite program.

8. Between 1992 and 1993, while at Hughes Aircraft Company, I designed and built a state-of-the-art, 800 megabit-per-second (Mbps) telecommunications modem for the NASA Lewis Research Center.

9. From 1997–2003, I worked at 8x8, Inc., starting as a DSP software engineer in 1997, becoming a manager in 1998, a director in 1999, and Vice President of Engineering in 2000 (managing a team of approx. 60 engineers in the company’s microelectronics group). I played a key role in developing several semiconductor products used worldwide in multimedia and communications devices, mainly for video conferencing systems and Internet Protocol (“IP”) telephones. Some of these semiconductor products were in production more than ten years.

10. From 2003–2007, I was a Systems Architect and Engineering Program Manager at Texas Instruments in the Digital Subscriber Line (“DSL”) product business unit. At Texas Instruments, I was directly involved in the architecture, design, development and production of multicarrier DSL modem products. My work specifically included architecting a multicarrier DSL semiconductor and software product and managing all aspects of its development from inception to production.

11. My Ph.D. dissertation and related publications are in the fields of statistical signal processing and communication systems, and more specifically in the area of impulsive noise modeling for communication systems.

12. In 1995 I was the instructor for the graduate Statistical Signal Processing class (EE278) in the Electrical Engineering department at Stanford University. Prior to teaching this class, I was a teaching assistant for ten different classes in signal processing and radio frequency electronics at Stanford.

13. I have developed, and managed the development of, several successful semiconductor, software and systems products in the communications and multimedia fields. These products are listed in the attached *curriculum vitae*.

II. INFORMATION AND MATERIALS CONSIDERED

14. In the course of preparing this expert declaration I have considered the Family 6 patents, their respective file histories, both parties' proposed constructions for certain terms appearing in those patents, as well as any additional documents I cite in this declaration, such as dictionaries.

15. I have relied also on my own personal knowledge relating to the technology at issue, including my education, work experience, and training in this field of art. I have also relied on my understanding of the training, knowledge and skill level of a person of ordinary skill in the art at the time of the invention of the Family 6 patents.

III. PERSON OF ORDINARY SKILL IN THE ART

16. I understand that a person of ordinary skill in the art is considered to have the normal skills and knowledge of a person in a certain technical field, as of the time of the invention at issue. I understand that factors that may be considered in determining the level of ordinary skill in the art include: (1) the education level of the inventor; (2) the types of problems encountered in the art; (3) the prior art solutions to those problems; (4) the rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the education level of active workers in the field.

17. With respect to the Family 6 patents, a person of skill in the art would have an electrical engineering background and experience in the design of multicarrier communication systems, such as those employing orthogonal frequency division multiplexing ("OFDM") or

discrete multitone (“DMT”) modulation. More particularly, a person of skill in the art would be a person with a bachelor’s degree in electrical engineering (or a similar technical degree or equivalent work experience) and at least 3 years of experience working with such multicarrier communication systems.

18. I have 18 years of combined industrial and academic experience in the architecture, design, development, testing and production of communication systems. Furthermore, I have worked directly in the field of multicarrier communication systems, including product design and development, with many engineers meeting the standard defined in the previous paragraph for a person of skill in the art. Therefore, I certainly understand how a person of ordinary skill in the art would interpret or understand the claims of the Family 6 patents in light of the specifications and file histories.

IV. BACKGROUND OF THE TECHNOLOGY

A. Data Communications

19. Data communication between a first and second transceiver is effected by communicating one or more digital bits that represent, for example, data bits from a bitstream. The data bits are communicated over a communication medium such as a phone line. Data communication systems, for example, systems described by the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards (*e.g.*, ITU-T Recommendation G.992.3 (07/2002) (hereinafter “G.992.3”) and ITU-T Recommendation G.993.1 (06/2004) (hereinafter “G.993.1”)), which are incorporated into the Family 6 Patents by reference, use a multicarrier communication scheme. *See* A10 (’835 patent) at 1:53–55¹; A96–96 (G.992.3), § 1 (“Scope”); A97 § 5 (“Reference Models”). *Id.* In a multicarrier communication scheme, such as the DMT

¹ The ’835 and ’162 patents share a common disclosure. For brevity, this document cites only to the ’835 patent.

schema used in G.992.3, for example, groups of hundreds or thousands of bits may be communicated at one time.

B. Interleaving and Error Correction

20. The ‘835 patent describes:

Communications systems often operate in environments that produce impulse noise. Impulse noise is a short-term burst of noise that is higher than the normal noise that typically exists in a communication channel. For example, DSL systems operate on telephone lines and experience impulse noise from many external sources including telephones, AM radio, HAM radio, other DSL services on the same line or in the same bundle, other equipment in the home, etc. It is standard practice for communications systems to use interleaving in combination with Forward Error Correction (FEC) to correct the errors caused by impulse noise.

A10 (‘835 patent) at 1:27-37.

21. Interleaving, as explained in the above-quoted passage from the ‘835 patent, is a technique used to reduce errors in data communications caused by impulse noise (*i.e.*, short-term bursts of noise). Interleaving (or an interleaver) takes an input byte stream (a byte is typically a group of eight bits) and then re-arranges and/or delays the input bytes according to a known pattern to produce an output byte stream. *See* A118 (G.992.3), § 7.7.1.5 (including Table 7-13). An interleaver is used in a transmitter, and a corresponding de-interleaver is used in a corresponding receiver to put the bytes back in the same order as the original input byte stream. *Id.*

22. As further explained in the above-quoted passage from the ‘835 patent, interleaving is typically used in conjunction with a Forward Error Correction (“FEC”) method. FEC methods, such as “Reed-Solomon encoding,” encode input data bits into “codewords” that include the input data bits and additional “parity” bits. *See* A117–18 (G.992.3) at § 7.7.1.4, and A129–130 (G.993.1) at § 8.3. The FEC encoding results in an FEC output byte stream. *Id.* The

FEC output byte stream is then interleaved by the interleaver. *See* A106–107 (G.992.3) at § 7.4, and A128–29 (G.993.1) at § 8.1. The process of interleaving, *i.e.*, rearranging and/or delaying the adjacent bytes of the FEC output byte stream, before transmitting helps reduce bit errors due to impulse noise when the byte stream is received, deinterleaved, and decoded at the receiver. *See* A117–18 (G.992.3) at § 7.7.1.5 (describing “[t]o spread the Reed-Solomon codeword and therefore reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved”), and A130 (G.993.1) at § 8.4.1 (describing “[i]nterleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords”). For purposes of the disputed Family 6 claim constructions, it is not important to understand exactly how error correction is accomplished; rather, it is enough to know that the goal of interleaving is to rearrange the order of the FEC byte stream so that, when the byte stream is transmitted, previously-adjacent bytes are spread out over time.

23. DSL communication systems such as those described as exemplary embodiments in the Family 6 patents use a type of interleaver called a “convolutional interleaver.” *See* A118 (G.992.3) at § 7.7.1.5 (describing “spread[ing] the Reed-Solomon codeword and therefore reduc[ing] the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved”). A convolutional interleaver partitions its input into “blocks” of bytes, and its operation is defined by two parameters: the block length “I” and the depth parameter “D.” *See, e.g.,* A130 (G.993.1) at § 8.4.1. Given these two parameters, a convolutional interleaver works as follows: (1) the first byte of each input block is not delayed in the output, (2) the second byte of each input block is delayed by D-1 bytes, such that it is spaced D bytes from the first byte in the output, (4) the third byte is delayed such that it is spaced

D bytes from the second byte in the output, (5) and so on, spreading out the bytes of an input block until such that any two bytes in the block that were once adjacent to each other are now D bytes away from each other. This results in the last byte of the block (the “I” byte) being delayed $(I-1)*(D-1)$ from its original position. This operation is depicted in Figure 1 below, using example parameters $I=5$ and $D=3$:

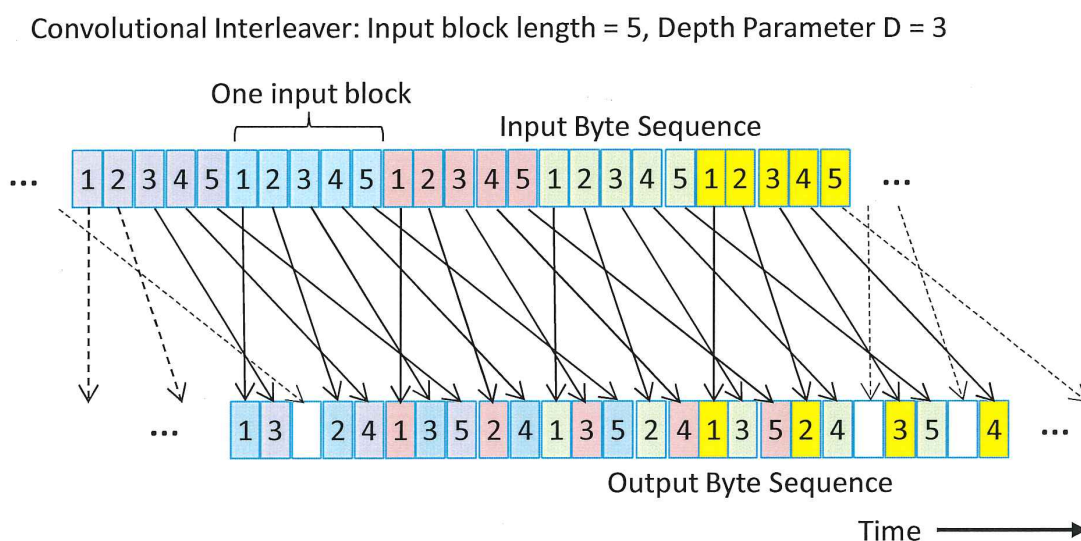


Figure 1: Convolutional interleaver with $I = 5$ and $D = 3$.

24. Figure 1 can be described as follows: (1) the position indexing 1, 2, 3, 4 and 5 of the bytes in each color-coded input block are for the convenience of the reader, and do not necessarily refer to specific data values; (2) the input byte stream and output byte streams are continuous, as represented by ellipses, (3) arrows coming from input bytes not shown or going to output bytes not shown are dashed instead of solid.

25. With reference to the input block shown in Figure 1 above that is colored blue and annotated “One input block,” its first byte “1” is not delayed and, therefore, its arrow is straight

down to the same blue “1” in the output sequence. However, its byte “2” is delayed by 2 spaces (or byte positions) from where it would otherwise have been placed in the output sequence (because $(2-1)*(D-1) = (2-1)*(3-1) = 2$), thus delaying it 2 byte positions from where it would have been and placing it three away from the blue “1” in the output. Likewise, the blue “3,” “4” and “5” are delayed by 4, 6 and 8 respectively from where they would have been had the sequence not been interleaved, thereby spacing the blue input bytes by three from each other in the output (because the interleaver depth parameter D is three in this example).

26. The description above of how a convolutional interleaver works is supported by the DSL standards themselves. For example, § 7.7.1.5 of G.992.3 states:

Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters D_p and the derived parameter $N_{FEC,p}$):

Each of the $N_{FEC,p}$ octets $B_0, B_1, \dots, B_{N_{FEC,p}-1}$ in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet B_i (with index i) is delayed by $(D_p - 1) \times i$ octets, where D_p is the interleaver depth.

An example for $N_{FEC,p} = 5, D_p = 2$ is shown in Table 7-13, where B_i^j denotes the i -th octet of the j -th FEC Output Data Frame.

Table 7-13/G.992.3 – Convolutional interleaving example for $N_{FEC,p} = 5, D_p = 2$

Interleaver input	B_0^j	B_1^j	B_2^j	B_3^j	B_4^j	B_0^{j+1}	B_1^{j+1}	B_2^{j+1}	B_3^{j+1}	B_4^{j+1}
Interleaver output	B_0^j	B_3^{j-1}	B_1^j	B_4^{j-1}	B_2^j	B_0^{j+1}	B_3^j	B_1^{j+1}	B_4^j	B_2^{j+1}

27. The “octets” discussed in the passage from § 7.7.1.5 of G.992.3 above correspond to the bytes of the interleaver input block, and the “octet index” corresponds to the byte position. Table 7-13/G.992.3 shows the same relationship between interleaver input byte positions and

interleaver output byte positions as discussed with respect to the colorized Figure 1 except that an interleaver depth of 2 (for Table 7-13/G.992.3) is used instead of 3 (for Figure 1).

28. Figure 2 below shows an interleaver input and partial views of interleaver outputs using an interleaver block size of $I=7$ and an interleaver depth of $D=5$ to illustrate alternative ways of thinking about interleaver depth:

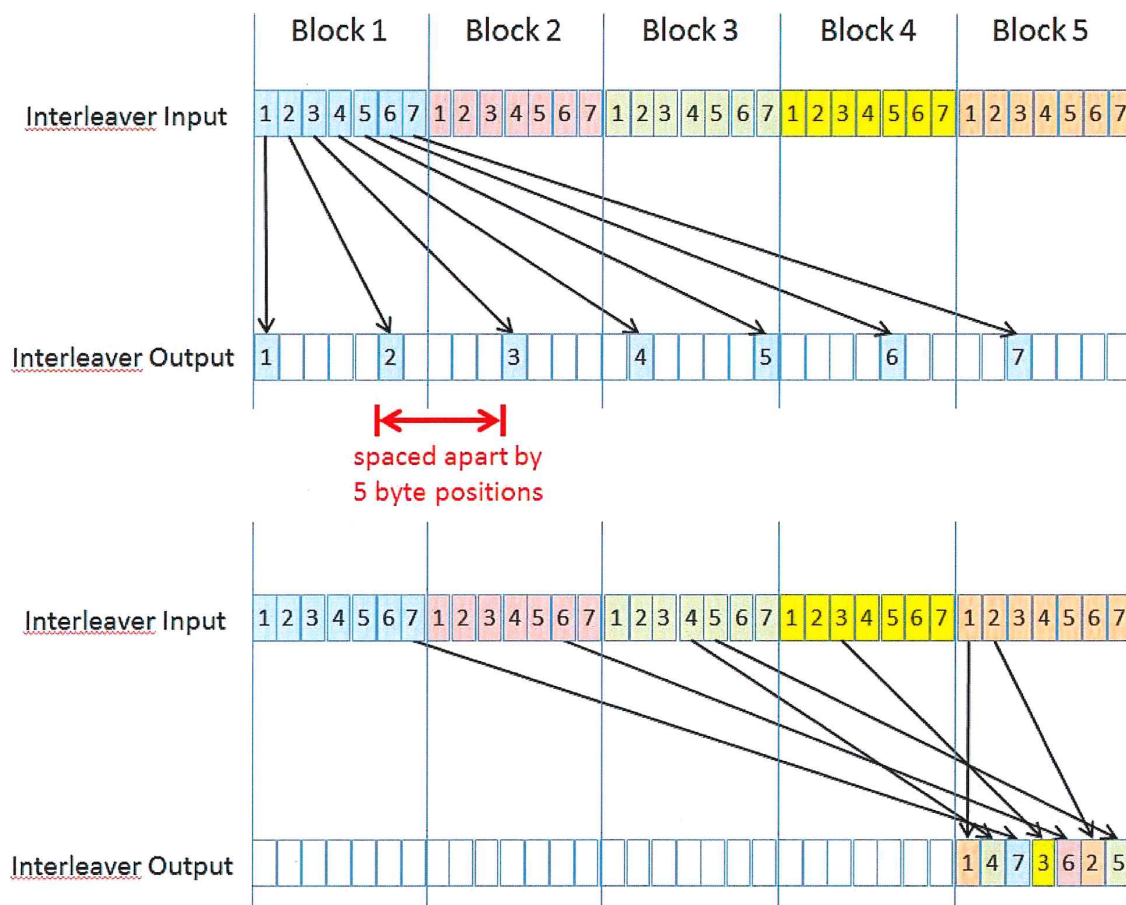


Figure 2: Convolutional interleaver with $I = 7$ and $D = 5$.

At the top of Figure 2, only the outputs corresponding to the blue input block are shown. Similar to the discussion for Figure 1, the first byte “1” of the blue block is not delayed. The second byte “2” of the blue block is delayed by four byte positions from where it would have been per the equation $(2-1)*(D-1) = (2-1)*(5-1) = 4$, thus spacing it five byte positions from the blue “1” in the output. Likewise, blue bytes 3, 4, 5, 6, and 7 of the blue block are delayed by 8, 12, 16, 20, and 24 byte position, respectively, from where they would have been without interleaving. The result is that in the interleaver output each byte from the blue block is now spaced five byte positions from its nearest original neighbor byte(s) from the blue block. This illustrates one way of thinking of interleaver depth, *i.e.*, that the interleaver depth will determine the spacing in the output of bytes from the same block that were adjacent in the input.

29. In the top of Figure 2, the blue “7” in the output is delayed by 24 byte positions from its original position in the sequence. The 24 corresponds to $(I-1)*(D-1) = 6 * 4 = 24$, and is the maximum delay of the interleaver. *See* A130 (G.993.1) § 8.4.1 (stating that the absolute read-to-write delay Δ_j equals $(D-1) * j$ where $j = 0, 1, 2, \dots, I-1$.) Since the first byte of the input block is not delayed, and the last byte of the input block is delayed by $(I-1)*(D-1)$, one input block spans $I + ((I-1)*(D-1))$ bytes in the output. This term simplifies to $(I*D) - D + 1$, which can be approximated for $I > D$ as simply $I * D$, meaning that the interleaver “spreads out” each input block to span D input block lengths in the interleaver output. This relationship can be seen in the in the top of Figure 2: the blue input block is now spread over the five output blocks, Block 1 through Block 5. This illustrates another way of thinking of interleaver depth, *i.e.*, that the interleaver depth will determine the number of output blocks over which an input block will be spread.

30. The bottom of Figure 2 illustrates yet another way of thinking of interleaver depth. As shown, output Block 5 includes at least one byte from each of input Blocks 1-5. So interleaver depth can also be thought of as the number of input blocks that have at least one byte in an output block.

31. In a DSL transmitter, the interleave operation typically follows the FEC operation, *i.e.*, the interleaver's input consists of FEC output codewords. *See, e.g.*, A117–18 (G.992.3) §§ 7.7.1.4 and 7.7.1.5 (including Figure 7-6, reproduced below in this declaration); A128–29 (G.993.1) § 8.1 (including Figure 8-1, reproduced below in this declaration). For some DSL standards, the input block size of the interleaver may be equal to the FEC output codeword size. *See* A117–18 (G.992.3) §§ 7.7.1.4 and 7.7.1.5. In this case, with reference to the paragraph above, the interleaver depth parameter *D* may be viewed as approximately the number of FEC codeword lengths over which any one input FEC codeword is spread out in the output, which is also an approximation of the delay—in units of FEC codewords added by the interleaver. A10 ('835 Patent) at 2:14. However, the interleaver input block size does not have to be equal to the FEC output codeword size. In G.993.1, for example, a standard referenced by the Family 6 Patents, the interleaver block length *I* can be a number that divides into the FEC codeword length. *See* A130 (G.993.1) at § 8.4.1 (describing “[t]he block length *I* shall divide the RS codeword length *N* (*i.e.*, *N* shall be an integer multiple of *I*)”). Put another way, a G.993.1 system can be configured such that one FEC codeword is divided into an integer number of interleaver input blocks. *Id.* Irrespective, however, of whether an input interleaver block is the size of a codeword or the size of a fraction of a codeword, the interleaver depth will still determine how bytes of an input block are spaced in the interleaver output in the manner described above. By way of example, if the interleaver block size is $\frac{1}{4}$ of a codeword and *D*=5,

then one codeword in the input (*i.e.* four interleaver blocks) will span only 8/4 codewords (or 2 codewords) in the output (since four interleaver blocks in the input will span $3 + 5 = 8$ interleaver block lengths in the output, which is 8/4 codewords.) In this case, the interleaver parameter $D=5$ is not “a number of codewords” and, thus, one of skill in the art would understand that interleaver depth is not necessarily expressed in codewords as Defendants’ claim constructions require.

32. In view of the foregoing, it would be incorrect to interpret the interleaver parameter D as being an interleaver depth “in number of codewords.” This is particularly so because the claims themselves do not require that the interleaver depth be expressed in codewords.

V. THE FAMILY 6 PATENTS

33. The Family 6 patents provide a novel solution for impulse noise protection adaptation. *See* A10 (’835 patent) at 1:20–25, A13 8:4–9:18, and A3–A8 (Figures 1, 3, 4 and 6).

34. As described in the Family 6 patents and in Section IV above, communication systems may use interleaving in conjunction with FEC coding to counter the effects of impulse noise. *See* A10 (’835 patent), 1:27–37 (stating “[i]t is standard practice for communications systems to use interleaving in combination with Forward Error Correction (FEC) to correct the errors caused by impulse noise”). Figure 7-6 from the G.992.3 standard, reproduced below, depicts the interleaver following the FEC block in the data processing operations of a transmitter.

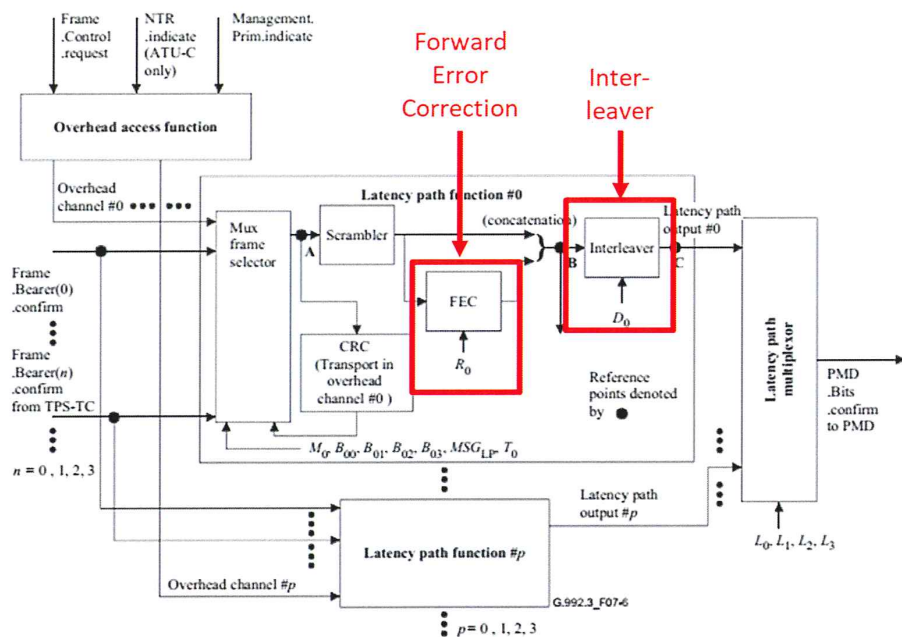


Figure 7-6/G.992.3 – Block diagram of transmit PMS-TC function

A106–107 (G.992.3) at § 7.4 and Figure 7-6 (annotated with red boxes around the FEC and Interleaver blocks, and corresponding red arrows and labels). Likewise, Figure 8-1 from the G.993.1 standard, reproduced below, also depicts the interleaver following the FEC block in the data processing operations of a transmitter.

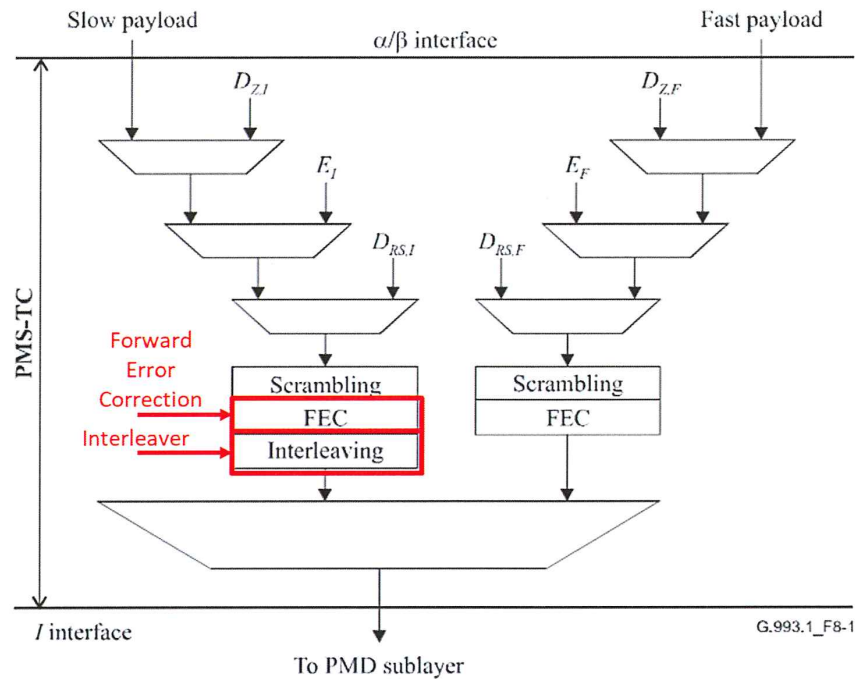


Figure 8-1/G.993.1 – Diagram of PMS-TC sublayer

A128–29 (G.993.1) at § 8.1, Figure 8-1 (annotated with red boxes around the FEC and Interleaving blocks, and corresponding red arrows and labels). The corresponding receiver has a de-interleaver followed by an FEC decoder. *See* A128–32 (G.993.1) at §§ 8, 8.4 and 8.4.2; A13 ('835 patent) at 8:8–17.

35. Parameters associated with FEC and interleaving include an FEC input block length in octets (*i.e.*, bytes) K , a number of added FEC redundancy octets R , the FEC codeword size $N = K + R$, and an interleaver depth D . *See* A10 ('835 patent) 2:22–25, A117–18 (G.992.3) §§ 7.7.1.4 and 7.7.1.5. As described in Section IV.B above, the G.993.1 standard also defines a parameter $q = N/I$ since interleaver block length I may be different from the FEC codeword size N . *See* A128–32 (G.993.1) at §§ 8 and 8.4.2, and Table 8-1. The '835 patent refers to FEC and

Interleaving Parameters as “FIP” (also referred herein as “FIPs” or “FIP parameters”). *See* A10 (’835 patent), 2:22–25.

36. The Family 6 patents describe a problem wherein some communication systems, *e.g.* DSL systems, do not properly characterize impulse noise and therefore may not choose FIP parameters that enable a sufficiently low bit error rate. *See id.* at 1:45–62.

37. The Family 6 patents explain that prior-art solutions to this problem are ad-hoc, time consuming and often result in FIP values that provide more impulse noise protection than needed, resulting in lower data rates and longer latency than needed. *See* A10–11 at 2:22–3:16.

38. The Family 6 patents describe inventive features that improve a communication system’s ability to deliver a sufficiently low error rate in the presence of impulse noise without (a) compromising high data rate and low latency performance more than necessary, or (b) requiring repeated and lengthy re-initialization procedures that interrupt steady-state data transmission (*i.e.*, “Showtime”). *Id.* The ’835 patent describes:

In accordance with one particular aspect of this invention, the system can transition from one FIP setting to another FIP setting without going through the startup initialization procedure such as the startup initialization sequence utilized in traditional xDSL systems. For example, an xDSL system that implements the systems and methods described herein could start using an FIP setting of (N=255, K=247, R=8, D=64) and then transition to an FIP setting of (N=255, K =239, R = 16, D=64) without re-executing the startup initialization procedure.

A11 at 3:37–47.

39. The Family 6 invention that enables updating FIPs during Showtime is depicted in Figures 3 and 6 and explained by the associated written description. The ’835 patent describes:

FIG. 3 outlines an exemplary method for performing impulse noise protection adaptation during Showtime according to this invention. In

particular, control begins in step S300 and continues to step S310. In step S310, traditional DSL initialization occurs. Next, in step S320, Showtime is entered between the two modems using the first FIP setting that was determined during the initialization in step S310. Then, in step S330, a determination is made whether bit errors are occurring using the first FIP setting. If bit errors are not occurring, control continues to step S340 where the control sequence ends. Otherwise, control jumps to step S350.

In step S350, a determination is made that an increase of the INP setting is required that requires modification of the FIP parameters. Next, in step S360, updated INP parameter is determined and a message forwarded to the receiver specifying the new INP setting. Then, in step S370, the receiver forwards to the transmitter updated FIP parameters for the new impulse noise protection requirements. Control then continues to step S380.

In step S380, the transmitter and receiver transition to using the updated INP parameters at a synchronization point. Next, in step S390 Showtime operation continues. Control then continues back to step S330.

A18 at 18:3–26, and A5 (Fig. 3).

FIG. 6 illustrates an exemplary method of synchronization using a flag signal according to this invention. In particular, control begins in step S600 and continues to step S610. In step S610, the modems enter Showtime using the first FIP parameters. Next, in step S620, a message is exchanged indicating the new FIP settings. Then, in step S630, the transmitter forwards to the receiver a flag signal indicating when the new FIP settings are to be used.

At step S640, and at a predefined change time following the transmission of the flag signal, the transmitter begins transmission using the new FIP parameters. Next, at step S650, at the predefined change time following the reception of the flag signal, the receiver commences reception utilizing the new FIP parameters. Control then continues to step S660 where Showtime communication continues with the control sequence ending at step S670.

A19 at 19:15–30, and A8 (Fig. 6).

40. Therefore the Family 6 patents teach that a flag signal is sent from a receiving transceiver to a transmitting transceiver to synchronize the transition to new FIP settings is to occur. *Id.* At a predefined change time following the transmission of the flag signal, the transmitter begins transmission using the new FIP parameters. *Id.* In one embodiment that performs “synchronization using a flag signal, the receiver and transmitter would start using updated FEC and interleaving parameters on a pre-defined FEC codeword boundary following the sync flag.” A15 ('835 Patent) at 12:8–11.

VI. DISPUTED CLAIM TERMS

A. “transceiver”

41. I understand that this term has been construed as “a communications device capable of transmitting and receiving data wherein the transmitter portion and receiver portion share at least some common circuitry.” I agree with this construction.

42. A “transceiver” is a well-understood term of art and under the generally accepted definition, a transceiver is capable of transmitting and receiving and the transmitting and receiving functions are implemented using at least some common circuitry. *See* A85 (Merriam Webster’s Collegiate Dictionary at p. 1253 (“a radio transmitter-receiver that uses many of the same components for both transmission and reception”)); and A81 (IEEE Standard Dictionary of Electrical and Electronic Terms at p. 1028 (“The combination of radio transmitting and receiving equipment in a common housing, . . . , and employing common circuit components for both transmitting and receiving”)). A person of skill in the art would understand that a digital signal processor (“DSP”), a memory or storage component, a transmit/receive switch, an antenna and/or other components could be shared, for example, between the transmitter and the receiver.

43. In contrast, the broad definition of “transceiver” proposed by Defendants could incorrectly be interpreted to include those types of designs where the transmitter is isolated from, and functionally unrelated to, the receiver (*e.g.*, communicating using an entirely different communication scheme over a different medium). This is not contemplated by the patents-in-suit or the dictionaries cited above. Any construction of “transceiver” that could be interpreted as including those types of designs where the transmitter is isolated from the receiver is incorrect.

44. Based on the foregoing, “transceiver” should be construed to mean a “communications device capable of transmitting and receiving data wherein the transmitting portion and receiving portion share at least some common circuitry.”

B. “flag signal” for ’835 patent, claims 8 and 10

45. Consistent with the specification and the understanding of a person of ordinary skill in the art, a flag signal is an inverted sync symbol or sync flag used to synchronize the transition to using updated forward error correction and/or interleaving parameters (*i.e.*, FIP settings). A15 (’835 patent) at 12:3–5.

46. The Family 6 patents describe several ways the receiver and transmitter can synchronize the modification of the FIP settings such that both the transmitter and receiver are coordinated to start using the updated settings at the same instant in time, including: 1) using a particular FEC codeword count that is communicated in a message from one transceiver to the other, or 2) using a flag signal. A15 (’835 patent) at 11:4–9. The synchronization using a communicated FEC codeword count is described as being message based, because “[p]rior to the transition point, the transmitter 300 or receiver 200 in cooperation with the message module 290

and message module 330, would send a message indicating the FEC codeword count value on which the FIP parameters will be updated.” *Id.* at 11:16–20.

47. However, the synchronization based on a flag signal is not based on a control or overhead message such as one using message module 290. Instead, the Family 6 patents describe the flag signal as being a “sync flag” and as being a “flag or marker signal that is similar to that used in the ADSL2 G.992.3 ORL [sic, OLR] protocol.” *Id.* at 12:3–5. A flag or marker signal similar to that used in the ADSL2 G.992.3 OLR protocol is an inverted sync signal, also described as a sync flag. *See* A92–A93 (G.992.3), §§ 8.7 and 8.7.3.

48. Defendants’ proposed construction is inconsistent with the specification and with how such term is generally used in the art. Specifically, the Family 6 patents describe using a flag signal to synchronize the switching between one FIP setting and another, but the “flag signal” does not by itself contain information “indicating when updated settings are to be used” (as proposed by the Defendants). A15 (’825 patent) at 12:4–11, 25–31.

49. Specifically, in embodiments of the Family 6 patents that use a flag signal, a transceiver upon receipt of the flag signal would determine when to begin using an updated FIP setting. *Id.* Such information is not contained in the flag signal itself; rather, the transceivers know when to use the updated FIP settings based on the timing of the transmission/receipt of the flag signal and on a pre-defined protocol known to both the transmitter and receiver. *Id.* To this point, the full claim term in which “flag signal” is found in independent claim 8 recites “the switching occurs on a pre-defined forward error correction codeword boundary following the flag signal.” A20 (’835 patent) at claim 8. Thus, the flag signal is used to synchronize the switch, but does not itself contain information “indicating when” the switch will occur.

50. Therefore, for claims 8 and 10 of the '835 patent, which include the claim limitation “switch to using for transmission, a second FIP setting,” the term “flag symbol” should be construed as “inverted sync symbol or sync flag used to synchronize the switch to using an updated FIP setting.”

C. “flag signal” for '162 patent, claims 8 and 9

51. As described for Section VI.B above, in the context of the Family 6 Patents, a flag signal is an inverted sync symbol or sync flag used to synchronize the transition to using updated framing and/or interleaving parameters. Therefore, for the '162 patent claims 8 and 9, which include the claim limitation “change to transmitting using a second interleaver parameter value that is different than the first interleaver parameter value,” the term “flag symbol” should be construed as “inverted sync symbol or sync flag used to synchronize the change to using an updated interleaver parameter value.”

52. Defendant’s proposed construction is inconsistent with the specification and with how such term is generally used in the art for the same reasons as discussed in Section VI.B. above.

D. “interleaver parameter value”

53. As described in the interleaver overview Section IV.B above, a person of skill in the art would understand that the Family 6 Patents contemplate a convolutional interleaver and deinterleaver. As also described in that section, it would be incorrect to require that the interleaver parameter value be expressed in terms of a number of codewords because that interpretation does not apply when the interleaver input block length is not equal to the length of a FEC codeword, as is the case for the G.993.1 VDSL standard incorporated by reference into the Family 6 patents. Instead, “interleaver parameter value” is simply a “numerical value of an

interleaver depth parameter.” Furthermore, the manner in which the interleaver parameter value is used in each of the example DSL standards that are incorporated by reference in the Family 6 patents does not require the interleaver depth to be expressed in any particular units, much less in codewords. Rather, interleaver depth is simply a number that defines, per the DSL standards, how the byte position of bytes of the input blocks will be delayed (and therefore, spaced apart) in the interleaved output.

E. “FIP setting”

54. An “FIP setting,” or “FEC and Interleaving Parameters setting,” is a set including at least one forward error correction parameter value and at least one interleaver parameter value.

55. Defendants’ use of the phrase “interleaver depth in number of codewords” in their construction is incorrect for the reason described in Section VI.D (“interleaver parameter value”) above: it is not a requirement of the claim language and such a narrow interpretation does not apply when the interleaver input block length is not equal to the length of a FEC codeword, as is the case for the G.993.1 VDSL standard incorporated by reference into the Family 6 patents. A130-A132 (G.993.1) at §§ 8.4.1 and 8.4.2.

F. “FIP value”

56. An “FIP value,” or “FEC and Interleaving Parameters value,” is a numerical value of a forward error correction parameter or numerical value of an interleaver parameter.

57. Defendants’ use of the phrase “interleaver depth in number of codewords” in their construction is incorrect for the reason described in Section VI.D (“interleaver parameter value”) above: it is not a requirement of the claim language and such a narrow interpretation does not apply when the interleaver input block length is not equal to the length of a FEC codeword, as is

the case for the G.993.1 VDSL standard incorporated by reference into the Family 6 patents. A130-A132 (G.993.1) at §§ 8.4.1 and 8.4.2.

G. “the switching occurs on a pre-defined forward error correction codeword boundary”

58. The term “switching occurs on a pre-defined forward error correction codeword boundary” has a definite meaning to a person of skill in the art implementing the ’835 patent.

59. For the incorporated DSL standards, *i.e.*, the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards, both the transmitter and receiver are continuously aligned and synchronized during normal data operation (*i.e.*, “showtime”) with respect to FEC codeword boundaries; *i.e.*, each knows the exact boundary of each FEC codeword. *See* A_ (G.992.3) at §§ 7.6 and 7.7, A_ (G.993.1) at § 8.5. Since the same protocol is implemented in the transmitter and the receiver, and both know the position in the byte stream (and, therefore, location in time) of FEC codeword boundaries, codeword boundaries are known in advance to both transceivers. Thus, to require that “the switching occurs on a pre-defined forward error correction codeword boundary following the flag signal” is definite.

60. The ’835 patent explains “[f]or synchronization using a flag signal, the receiver and transmitter would start using updated FEC and interleaving parameters on a pre-defined FEC codeword boundary following the sync flag.” A15 (’835 patent) at 12:8–11. In this context, “pre-defined” means that both the transceiver and receiver independently know how to calculate an exact FEC codeword boundary transition point relative to an occurrence of the sync flag, based on a protocol defined by a communication standard document and/or by the designer of the transceiver devices.

61. Since the same protocol is implemented in the transmitter and the receiver, and both know the position in the byte stream (and, therefore, location in time) of FEC codeword

boundaries, codeword boundaries are known in advance to both transceivers. Thus, to require that “the switching occurs on a pre-defined forward error correction codeword boundary following the flag signal” is definite.

H. “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary”

62. The term “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary” has a definite meaning to a person of skill in the art implementing the ’162 Patent.

63. As describes in section VI.G above for the term “the switching occurs on a pre-defined forward error correction codeword boundary,” the same protocol is designed into the transmitter and the receiver, and both know the FEC codeword boundaries. Therefore, a person of skill in the art would have an unambiguous understanding of “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary following transmission of the flag signal.

64. I declare under penalty of perjury of the law of the United States that the foregoing is true and correct.

Date: September 20, 2017



Dr. Douglas A. Chrissan

Douglas A. Chrissan

dchrissan@gmail.com

(408) 823-9976

EMPLOYMENT:

Engineering and Intellectual Property Consultant	2011–present
Maxim Integrated Products, Sunnyvale, CA (MXIM) Engineering Director, Video Processing	2009–2011
Keystream Corporation, Mountain View, CA Vice President, Engineering	2009
Texas Instruments, Sunnyvale, CA (TXN) Program Engineering Manager, DSP Division Systems Architect, DSL	2004–2009 2003–2004
8x8, Inc, Santa Clara, CA (EGHT) VP Engineering, Netergy Microelectronics group Director, Signal Processing Algorithms Manager, Audio Algorithms Senior Software Engineer	2000-2003 1999 1998 1997
Hughes Aircraft Company (now Boeing) Masters Fellow and Member of Technical Staff	1988-1993

EDUCATION:

Ph.D., Electrical Engineering, Stanford University	1998
M.S.E.E., University of Southern California	1990
B.S.E.E., University of Southern California	1988

EXPERIENCE SUMMARY:

Twenty years of communications, multimedia and networking experience with expertise in communication systems, embedded software, Internet server and client software, mobile device software, audio/video processing and system-on-chip (SoC) ICs.

Engineering and Intellectual Property Consultant	2011–present
TQ Delta Expert Witness for TQ Delta (Plaintiff), Civil Actions 1:15-cv-00611-RGA through 1:15-cv-00616-RGA (deposed once for this case).	2016-present

Expert Witness for TQ Delta (Patent Owner), IPR2016-01006, -01007, -01008, -01009, -01160 and -01466 (deposed once for these cases).

Monster, Inc. 2014
Expert Consultant for Monster, Inc. (Defendant), Civil Action No. 13 Civ. 8229 (KBF) (S.D.N.Y.) (case settled)

Intellectual Ventures 2013-present
Expert Witness for IV (Plaintiff), Civil Actions 1:13-cv-00116-LY (deposed twice for this case), 1:13-cv-00118-LY (case settled) and 1:13-cv-00119-LY (case settled).
Expert Consultant for IV (Plaintiff), Civil Actions 1:12-cv-00193-LPS and 1:13-cv-01668-LPS through 1:13-cv-01672-LPS.

Wiffledan, Inc. (a.k.a. Vphoto, acquired by Hulu) 2013-2015
Designed and implemented image processing and computer vision algorithms on iOS devices for selecting the most appealing images from a video sequence. These algorithms are used in the *Vphoto* application, available as of 2014 from the Apple App store.

Cavium, Inc. 2011-2012
Managed customer engineering for WiFi-enabled wireless remote display receivers, including the Samsung AllShareCast wireless remote display dongle.

Maxim Integrated Products, Sunnyvale, CA (MXIM) 2009–2011
Engineering Director, Video Processing

Managed and developed the MAX64380 High-Definition H.264 Video Compression/Decompression Integrated Circuit from inception to production.

Managed and developed the MAX64180 High-Definition H.264 Video Compression/Decompression Integrated Circuit from inception to production. This product was used in many internet-connected camera designs for security cameras and TV webcams (cameras connected to Smart TVs).

Managed and developed the iZon camera for Stem, Inc. using the MAX64180. This first-generation, WiFi-connected security camera was available as the Stem iZon product in Apple stores as of 2011.

Managed the development of several Skype TV Webcam designs; these designs enable Smart TVs to run Skype natively as a videoconferencing application on the TV. The first of these designs, for Samsung, was the first Skype TV webcam product in the market (2010).

Keystream Corporation, Mountain View, CA
 Vice President, Engineering 2009
 (Company ceased operations in Dec. 2009)

Managed an agile team of Internet software engineers in the development and release of company's SmartAd platform, an ad delivery system to users' web browsers.

Implemented computer vision algorithms for detecting, tracking and classifying objects in videos.

Texas Instruments, Sunnyvale, CA (TXN)
 Program Engineering Manager, DSP Division 2004–2009
 Systems Architect, DSL 2003–2004

Managed the development and release to production of the UR8 Digital Subscriber Line (DSL) ADSL2/VDSL2 residential gateway product, including all hardware and software components. This work included substantial contributions to the architecture and design of TI's TNET7531 and TNET7530 multi-core, DSL transceiver integrated circuits and related software. UR8 was a \$20M+ product development including 100 engineers at multiple worldwide sites, with first silicon released to production and all software delivered on schedule.

Authored the TI White Paper "Uni-DSL™: One DSL for Universal Service" (see publications section below).

Managed the ~\$50M divestiture of TI's digital subscriber line IC products to Infineon, including IT, technology transfer, support, operations and product engineering.

8x8, Inc, Santa Clara, CA (EGHT)
 VP Engineering, Netergy Microelectronics group 2000-2003
 Director, Signal Processing Algorithms 1999
 Manager, Audio Algorithms 1998
 Senior Software Engineer 1997

Directed a team of silicon, software, hardware and applications engineers in the development of Voice-over-IP (VoIP) and Video-telephony software and semiconductor products.

Substantially contributed to the Vision Compression Processor EX (VCP-EX) integrated circuit, the Audacity-T2 Voice-Over Internet Protocol Processor integrated circuit and the Audacity-T2U Voice-Over Internet Protocol Processor integrated circuit. The Audacity-T2 and T2U were in production and used in designs worldwide for more than ten years.

Managed major software releases and hardware product lines, enabling company's OEM customers to develop VoIP products including the Ericsson DRG-22 Ethernet Residential Gateway, the Telsey Ethernet Residential Gateway and the D-Link DPH-100 IP Phone.

Architected the design and managed the development of a DSP core for audio/video processing. This DSP core was licensed by ST Microelectronics and led to a \$27M investment in the company by ST.

Managed company's IP portfolio of ~50 patents and patent applications.

Designed and managed and designed the implementation of the G.7xx ITU speech compression algorithms on four different DSP architectures

Hughes Aircraft Company

Masters Fellow and Member of Technical Staff

1988-1993

Designed and developed a digitally synthesized, bandwidth efficient 800 Mb/s modem under a NASA Lewis Research Center contract.

Provided pre-sales technical and design support for commercial satellite programs, including the Aussat (Australia) and Palapa (Indonesia) programs.

Designed communication payload circuits for the Milstar satellite program.

PUBLICATIONS:

Douglas Chrissan, "Uni-DSL™: One DSL for Universal Service," Texas Instruments White Paper, SPAY018, June 2004.

Chrissan, D. A., and A. C. Fraser-Smith, "A Clustering Poisson Model for Characterizing the Interarrival Times of Sferics," *Radio Science*, 38, 17-1 to 17-14, 2003.

Chrissan, D. A., and A. C. Fraser-Smith, "A Comparison of Low-Frequency Radio Noise Amplitude Probability Distribution Models," *Radio Science*, 35, 195-208, 2000.

Chrissan, D. A., "Statistical Analysis and Modeling of Low-Frequency Radio Noise and Improvement of Low-Frequency Communications," Final Technical Report D179-1, Space, Telecommunications and Radioscience Laboratory, Stanford University, ONR Grants N00014-92-J-1576 and N00014-93-1-1073, August 1998. (Ph.D. dissertation)

Chrissan, D. A., and A. C. Fraser-Smith, "Diurnal Variations of Globally Measured ELF/VLF Radio Noise," *Tech. Report D177-2*, Space, Telecommunications and Radioscience Laboratory, Stanford University, ONR Grants N00014-92-J-1576 and N00014-93-1-1073, July 1997.

Chrissan, D. A., and A. C. Fraser-Smith, "Seasonal Variations of Globally Measured ELF/VLF Radio Noise," *Tech. Report D177-1*, Space, Telecommunications and Radioscience Laboratory, Stanford University, ONR Grants N00014-92-J-1576 and N00014-93-1-1073, December 1996.

Chrissan, D. A., and A. C. Fraser-Smith, "Seasonal Variations of Globally-Measured ELF/VLF Radio Noise," *Radio Science*, 31, 1141-1152, 1996.

Chrissan, D. A., and A. C. Fraser-Smith, "Seasonal Variations of ELF/VLF Radio Noise at Arrival Heights, Antarctica" *Antarctic J.*, 30, 368-369, 1996.

PATENTS:

Douglas A. Chrissan and Rajarathinam G. Subramanian, "Varying pulse amplitude multi-pulse analysis speech processor and method," U.S. Patent 7272553, Sep 18, 2007.

Bryan R. Martin, Ian John Buckley, Philip Bednarz and Douglas A. Chrissan, "Voice-Over Internet Protocol Processor," U.S. Patent 7,120,143, Oct. 10, 2006.

HONORS and AWARDS:

Armed Forces Communications and Electronics Association Graduate Research Fellow, 1994

Hughes Aircraft Masters Fellow, 1988-1990

National Merit Scholar, 1984-1988

**UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

TQ DELTA, LLC, Plaintiff, v. 2WIRE, INC., Defendant.	Civil Action No. 13-cv-1835-RGA
TQ DELTA, LLC, Plaintiff, v. ZHONE TECHNOLOGIES, INC., Defendant.	Civil Action No. 13-cv-1836-RGA
TQ DELTA, LLC, Plaintiff, v. ZYSXEL COMMUNICATIONS, INC., and ZYSXEL COMMUNICATIONS CORPORATION, Defendants.	Civil Action No. 13-cv-2013-RGA
TQ DELTA, LLC, Plaintiff, v. ADTRAN, INC., Defendant.	Civil Action No. 14-cv-954-RGA
ADTRAN, INC., Plaintiff, v. TQ DELTA, LLC, Defendant.	Civil Action No. 15-cv-121-RGA

**DECLARATION OF DR. KRISTA S. JACOBSEN IN SUPPORT OF
DEFENDANTS' FAMILY 6 CLAIM CONSTRUCTION BRIEF**

I. INTRODUCTION

1. My name is Krista S. Jacobsen. I have been asked by Defendants to provide this declaration in connection with the above-captioned District Court actions. Specifically, I have been asked to opine on the level of ordinary skill in the art at the time of the alleged invention, the state of the art at the time of the alleged invention, and how a person having ordinary skill in the art would have interpreted certain terms in U.S. Patent Nos. 8,462,835 (“the ’835 patent”) and 8,594,162 (“the ’162 patent”) at the time of the alleged invention. I have also been asked to opine on what I have identified as deficiencies of TQ Delta’s Opening Claim Construction Brief for the Family 6 patents and Dr. Douglas Chrissan’s declaration in support of TQ Delta’s opening Family 6 claim construction brief.

II. BACKGROUND AND QUALIFICATIONS

2. I was awarded a Ph.D. in Electrical Engineering from Stanford University in 1996, and a Master’s Degree in Electrical Engineering in 1993, also from Stanford University. I also hold a Bachelor of Science Degree in Electrical Engineering, *summa cum laude*, from the University of Denver, which I received in 1991.

3. My Ph.D. research focused on technology for digital communications, including multicarrier modulation, discrete multi-tone (DMT) modulation, and orthogonal frequency division multiplexing (OFDM). My doctoral thesis topic was “Discrete Multi-Tone-Based Communications in the Reverse Channel of Hybrid Fiber-Coax Networks.” My research adviser was Dr. John M. Cioffi, who is known as the “father of DSL.”

4. I am also a co-editor of two books on digital subscriber line (DSL) technology and the author of several book chapters on DSL technology and standardization. In addition, I have authored and co-authored numerous articles on multicarrier communications.

5. I have over ten years of experience working in the development and standardization of DSL technologies, including those embodied in the ITU-T Recommendations involved in the above-captioned District Court actions. I am an inventor named on eleven patents solving issues presented by or related to multicarrier modulation. My experience includes work in DSL technologies and the DSL industry both before and after the purported priority date of the Family 6 patents.

6. I am also licensed to practice before the Patent and Trademark Office. I am currently an attorney who serves as an expert consultant and witness for patent litigation, and I provide patent prosecution and counseling services in multiple areas, including telecommunications.

7. A detailed curriculum vitae showing more of my credentials is attached to this declaration as Exhibit A.

III. COMPENSATION

8. I am being compensated for my time at the rate of \$400 per hour. This compensation is not contingent on my performance, the outcome of this matter, or any issues involved in or relating to this matter.

IV. DOCUMENTS AND OTHER MATERIALS RELIED UPON

9. In forming the opinions set forth in this declaration, I have reviewed the asserted Family 6 patents and their file histories, including provisional application No. 60/594,804 (“the ’804 provisional”). Additionally, I have considered my own experience and expertise concerning the knowledge of a person having ordinary skill in the relevant art during the timeframe of the claimed priority date of the Family 6 patents. I have reviewed information generally available to, and relied upon by, a person having ordinary skill at the time of the alleged invention. I have also reviewed TQ Delta’s opening Family 6 claim construction brief

and Dr. Chrissan's strikingly-similar declaration in support of TQ Delta's opening Family 6 claim construction brief.

10. I was told to assume the time of the alleged invention is March 3, 2004, the date on which the '804 provisional was filed.

V. LEGAL PRINCIPLES

11. Although I am licensed to practice law, I do not opine in this paper on any particular methodology for interpreting patent claims. My opinions are based on the meaning that a term would have had to a person having ordinary skill in the art in light of the specification and the prosecution history at the time of the filing of the '804 provisional on March 3, 2004. I am not opining in this paper as a patent expert; I applied the principles below, however, as a guide in formulating my opinions.

12. I am informed and understand that it is a basic principle of patent law that before the validity or infringement of a patent claim can be assessed, the claim language must be properly construed to determine its scope and meaning as understood by persons of ordinary skill in the art. I am informed and understand that the words of a patent claim are to be given the meaning that those words would have had to a person having ordinary skill in the art at the time of the invention in light of the specification and the prosecution history. This meaning must be ascertained from a reading of the patent documents, paying special attention to the language of the claims, the drawings, the written description, and the prosecution history of the Family 6 patents. I understand that an inventor may attribute special meanings to some terms by defining those terms or by otherwise incorporating such meanings in these documents.

13. I am informed and understand that for claim construction, intrinsic evidence is more important and considered more reliable than extrinsic evidence. Consequently, where possible, I have relied primarily on intrinsic evidence in forming my opinions. Where I found

the intrinsic evidence to be lacking, I relied on my experience as a person who worked in the DSL field at the time of the alleged invention, as well as DSL standards that would have been known to persons having ordinary skill in the art.

14. I am informed and understand that a patent applicant may, at the time of filing an application, include a statement incorporating by reference the contents of other documents. I understand that, to be effective, an incorporation by reference must use the root words “incorporate” and “reference” (*e.g.*, “incorporate by reference”), must identify with detailed particularity what specific material it incorporates, and must clearly indicate where that material is found in the incorporated documents.

15. I am informed and understand that the material in a patent application is either “essential material” or “non-essential material.” I understand that “essential material” is material that is necessary to describe the claimed invention, to provide an enabling disclosure of the claimed invention, or to describe the best mode of using the invention. I understand that “essential material” may not be incorporated by reference to non-patent publications.

16. I am informed and understand that a continuation patent application is a patent application with the identical written description, drawings, and priority date as the parent application, but different claims. I also understand that a continuation application cannot include new matter.

17. My methodology for determining the meaning of claim terms and phrases was first to study the Family 6 patents. In particular, I studied the claims themselves, the written description (which includes the background, the summary, and the detailed description of the invention), and the drawings. Next, I reviewed the file histories of the Family 6 patents, looking for any clarifications of or limitations that might be attached to claim terms. In some

circumstances, I looked at other documents, such as provisional applications to which applications claimed priority, or documents purportedly incorporated by reference or referenced in the Family 6 patents.

VI. LEVEL OF SKILL IN THE ART

18. I disagree with Dr. Chrissan's characterization of the level of skill a person having ordinary skill in the art would have had on March 3, 2004. In my experience, at that time, a person having ordinary skill in the art would have had a bachelor's degree in electrical or computer engineering and 5-6 years of experience, a Master's degree in electrical engineering and 2-3 years of experience, or a Ph.D. in electrical engineering with 1-2 years of experience.

VII. BACKGROUND OF THE TECHNOLOGY

A. The June 2004 Version of G.993.1 Was Not Available to the Public On the Filing Date of the Family 6 Patents' Ultimate Parent Application

19. Dr. Chrissan and TQ Delta both contend that the Family 6 patents incorporate by reference the version of ITU-T Recommendation G.993.1 dated June of 2004, and they use the disclosures in this document and other standards documents extensively in their explanation of the Family 6 patents' technology. TQ Delta Brief at 1-12; Chrissan Dec. at ¶ 19 ("systems described by . . . the VDSL series of ITU G.993.x standards (*e.g.*, . . . ITU-T Recommendation G.993.1 (06/2004) (hereinafter 'G.993.1')), which are incorporated into the Family 6 Patents by reference, use a multicarrier communication scheme"); *id.* at ¶¶ 21-31. Although the version of the G.993.1 Recommendation dated June 2004 was *approved* in June of 2004, it was not *published or released to the public* until October 24, 2005, well after the filing date (and the publication date) of Patent Cooperation Treaty (PCT) application number PCT/US2005/006842 ("the '842 PCT application"), which is the ultimate parent application of the '835 and '162

patents. See <http://www.itu.int/rec/T-REC-G.993.1-200406-I/en> (indicating G.993.1 (06/2004) was not posted until October 24, 2005).

20. Therefore, the June 2004 version of G.993.1 would not have been available to a person having ordinary skill in the art as of the filing date of the '842 PCT application. Consequently, Dr. Chrissan's and TQ Delta's reliance on the June 2004 version of G.993.1 to explain the technology of the Family 6 patents is inappropriate because that document was not publicly available on the Family 6 patents' priority date and would not have been used by a person having ordinary skill in the art at the time of the alleged invention to understand the meanings of any terms in the claims.

B. The Family 6 Incorporation-By-Reference Statements Do Not Adequately Identify to a Skilled Artisan the Standards TQ Delta and Dr. Chrissan Contend Were Incorporated By Reference

21. Both statements in the Family 6 patents that purport to incorporate by reference certain ADSL and VDSL standards would not have clearly identified to a person having ordinary skill in the art exactly which documents the skilled artisan should understand to be incorporated.

For example, one attempted incorporation-by-reference statement is:

The most common method for measuring the SNR is to calculate the mean-square error of the received signal based on a known transmitted signal, which is described in the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards, which are incorporated herein by reference in their entirety.

'835 patent at col. 1:50-55 (emphasis added). Assuming, as a person having ordinary skill would, that the "x" in "G.992.x" and "G.993.x" captures all possible values of "x," on the filing date of the '842 PCT application, "the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards" included 20 documents totaling 1,381 pages of material.¹

¹ Specifically, "the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards" available to the public on March 3, 2005 were:

22. By comparison, on the filing date of the continuation application that matured into the '835 patent, "the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards" included, in addition to the documents that existed on the '842 PCT application's filing date, an additional 20 documents totaling 966 additional pages of material.² Thus, a person having ordinary skill in the art on the filing date of the application that matured

ITU-T Recommendation G.992.1 (07/1999) (posted 2/27/2004): 256 pages;
 ITU-T Recommendation G.992.1 Corrigendum 1 (11/2001) (posted 7/18/2002): 6 pages;
 ITU-T Recommendation G.992.1 Amendment 1 (03/2003) (posted 3/9/2004): 112 pages;
 ITU-T Recommendation G.992.1 Annex H (10/2000) (posted 8/3/2001): 44 pages;
 ITU-T Recommendation G.992.2. (07/1999) (posted 5/31/2000): 179 pages;
 ITU-T Recommendation G.992.2 Amendment 1 (03/2003) (posted 10/29/2003) with 50 pages;
 ITU-T Recommendation G.992.2 Amendment 2 (10/2003) (posted 2/27/2004): 12 pages;
 ITU-T Recommendation G.992.3 (07/2002) (posted 11/11/2003): 312 pages;
 ITU-T Recommendation G.992.3 (2002) Corrigendum 1 (12/2003) (posted 3/3/2004): 12 pages;
 ITU-T Recommendation G.992.3 (2002) Amendment 1, Corrigendum 1 (02/2004) (posted 3/2/2004): 6 pages;
 ITU-T Recommendation G.992.3 (2002) Amendment 2 (04/2004) (posted 9/16/2004): 34 pages;
 ITU-T Recommendation G.992.3 (2002) Amendment 3 (06/2004) (posted 6/28/2004): 68 pages;
 ITU-T Recommendation G.992.3 (2002) Amendment 4 (06/2004) (posted 2/18/2005): 26 pages;
 ITU-T Recommendation G.992.4 (07/2002) (posted 2/27/2003): 24 pages;
 ITU-T Recommendation G.992.5 (05/2003) (posted 2/4/2004): 92 pages;
 ITU-T Recommendation G.992.5 (2003) Corrigendum 1 (04/2004) (posted 8/31/2004): 16 pages;
 ITU-T Recommendation G.992.5 (2003) Amendment 1 (04/04) (posted 9/29/2004): 24 pages;
 ITU-T Recommendation G.992.5 (2003) Amendment 2 (06/2004) (posted 2/14/2005): 12 pages;
 ITU-T Recommendation G.993.1 (11/2001) (posted 8/29/2002): 48 pages; and
 ITU-T Recommendation G.993.1 Amendment 1 (03/2003) (posted 12/3/2003): 48 pages.
² Specifically, additional "ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards" available to the public on April 28, 2010 were:
 ITU-T Recommendation G.992.3 Erratum 1 (06/2007) (posted 7/24/2007): 1 page;
 ITU-T Recommendation G.992.3 Amendment 1 (09/2005) (posted 4/4/2006): 32 pages;
 ITU-T Recommendation G.992.3 Amendment 2 (03/2006) (posted 6/16/2006): 12 pages;
 ITU-T Recommendation G.992.3 Amendment 3 (12/2006) (posted 7/31/2007): 22 pages;
 ITU-T Recommendation G.992.3 Amendment 4 (07/2007) (posted 7/30/2008): 20 pages;
 ITU-T Recommendation G.992.3 Amendment 5 (06/2008) (posted 7/10/2009): 34 pages;
 ITU-T Recommendation G.992.5 (01/2005) (posted 8/10/2005): 110 pages;
 ITU-T Recommendation G.992.5 Amendment 1 (07/2005) (posted 7/12/2007): 32 pages;
 ITU-T Recommendation G.992.5 Amendment 2 (06/2006) (posted 9/1/2006): 12 pages;
 ITU-T Recommendation G.992.5 Amendment 3 (12/2006) (posted 7/27/2007): 14 pages;
 ITU-T Recommendation G.992.5 Amendment 4 (07/2007) (posted 12/14/2007): 10 pages;
 ITU-T Recommendation G.992.5 Amendment 5 (06/2008) (posted 4/30/2009): 8 pages;
 ITU-T Recommendation G.993.1 (06/2004) (posted 10/24/2005): 228 pages;
 ITU-T Recommendation G.993.2 (02/2006) (posted 1/30/2007): 252 pages;
 ITU-T Recommendation G.993.2 (2006) Corrigendum 1 (12/2006) (posted 9/21/2007): 26 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 1 (04/2007) (posted 4/30/2008): 100 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 2 (02/2008) (posted 6/5/2009): 28 pages;
 ITU-T Recommendation G.993.2 (2006) Corrigendum 2 (07/2007) (posted 8/7/2007): 4 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 1 Corrigendum 1 (07/2007) (posted 8/7/2007): 9 pages; and
 ITU-T Recommendation G.993.2 (2006) Corrigendum 3 (06/2009) (posted 11/19/2009): 12 pages.

into the '835 patent would have concluded that the incorporation-by-reference statement identified a total of 40 documents and 2,347 pages.

23. And, on the filing date of the application that matured into the '162 patent, “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” included, in addition to the documents that existed on the '842 PCT application’s filing date and on the filing date of the application that matured into the '835 patent, an additional 27 documents totaling 1,782 additional pages of material.³ Thus, a person having ordinary skill in the art on the filing date of the application that matured into the '162 patent would have concluded that the incorporation-by-reference statement identified a total of 67 documents and 4,129 pages.

24. Table 1, below, summarizes the documents identified by “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” on the various application filing dates.

³ Specifically, additional “ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” available to the public on June 11, 2013 were:

ITU-T Recommendation G.992.3 (04/2009) (posted 11/23/2010): 700 pages;
 ITU-T Recommendation G.992.3 (2009) Corrigendum 1 (11/2009) (posted 5/11/2010): 8 pages;
 ITU-T Recommendation G.992.3 (2009) Amendment 1 (03/2010) (posted 10/29/2010): 8 pages;
 ITU-T Recommendation G.992.3 (2009) Amendment 2 (07/2010) (posted 9/28/2010): 8 pages;
 ITU-T Recommendation G.992.3 (2009) Amendment 3 (11/2010) (posted 2/15/2011): 20 pages;
 ITU-T Recommendation G.992.3 (2009) Corrigendum 2 (06/2011) (posted 10/7/2011): 8 pages;
 ITU-T Recommendation G.992.3 (2009) Amendment 4 (10/2011) (posted 4/13/2012): 12 pages;
 ITU-T Recommendation G.992.3 (2009) Amendment 5 (10/2012) (posted 3/1/2013): 10 pages;
 ITU-T Recommendation G.992.5 (01/2009) (posted 11/23/2010): 176 pages;
 ITU-T Recommendation G.992.5 (2009) Corrigendum 1 (11/2010) (posted 3/14/2011): 10 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 3 (08/2008) (posted 2/21/2011): 88 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 4 (01/2009) (posted 12/7/2010): 8 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 5 (04/2010) (posted 9/3/2010): 26 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 6 (11/2010) (posted 12/21/2011): 18 pages;
 ITU-T Recommendation G.993.2 (2006) Corrigendum 4 (04/2011) (posted 7/5/2011): 8 pages;
 ITU-T Recommendation G.993.2 (2006) Amendment 7 (06/2011) (posted 12/21/2011): 82 pages;
 ITU-T Recommendation G.993.2 (09/2012) (posted 6/28/2012): 376 pages;
 ITU-T Recommendation G.993.2 (2011) Erratum 1 (09/2012) (posted 10/1/2012): 2 pages;
 ITU-T Recommendation G.993.2 (2011) Corrigendum 1 (06/2012) (posted 11/26/2012): 24 pages;
 ITU-T Recommendation G.993.2 (2011) Amendment 2 (12/2012) (posted 5/29/2013): 48 pages;
 ITU-T Recommendation G.993.2 (2011) Amendment 3 (04/2013) (posted 5/30/2013): 26 pages;
 ITU-T Recommendation G.993.5 (04/2010) (posted 9/7/2010): 80 pages;
 ITU-T Recommendation G.993.5 Corrigendum 1 (06/2011) (posted 11/25/2011): 10 pages;
 ITU-T Recommendation G.993.5 Amendment 1 (12/2011) (posted 5/4/2012): 18 pages; and
 ITU-T Recommendation G.993.5 Corrigendum 2 (06/2012) (posted 12/6/2012): 8 pages.

Table 1: Set of documents identified by “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” grows as time passes.

Date	Significance of date	Number of documents included in “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards”	Number of pages included in “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards”
March 3, 2005	Filing date of '842 PCT application	20	1,381
April 28, 2010	Filing date of '835 patent's application	40	2,347
June 11, 2013	Filing date of '162 patent's application	67	4,129

25. In my opinion, because the set of documents identified by and included in “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” shifts as time passes, a person having ordinary skill in the art would not be able to identify accurately exactly what documents were purportedly incorporated by reference in either of the asserted Family 6 patents. Stated another way, the set of documents a skilled artisan would consider to be included in “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” would depend on the date on which the skilled artisan set out to identify those documents. And because the ITU-T’s work on “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” continued after the filing date of the '162 patent, the set of documents included in “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” today differs from the sets that existed on the priority or filing dates of the Family 6 patents.

26. Even the statement purporting to incorporate by reference “the ADSL2 Standard G.992.3” ('835 patent at col. 1:65-67) would not have clearly identified to a skilled artisan

exactly which documents were incorporated by reference in the '835 patent or the '162 patent.

On the filing date of the '842 PCT application, “the ADSL2 Standard G.992.3” consisted of six documents totaling 458 pages. On the filing date of the application that matured into the '835 patent, “the ADSL2 Standard G.992.3” included six *additional* documents, leading to a total of 12 documents and 579 pages. And on the filing date of the application that matured into the '162 patent, “the ADSL2 Standard G.992.3” included an additional eight documents, including a new version of the standard, leading to a total of 20 documents and 1,353 pages.

27. Table 2, below, summarizes the documents identified by “the ADSL2 Standard G.992.3” on the various application filing dates.

Table 2: Set of documents identified by “the ADSL2 Standard G.992.3” grows as time passes.

Date	Significance of date	Number of documents included in “the ADSL2 Standard G.992.3”	Number of pages included in “the ADSL2 Standard G.992.3”
March 3, 2005	Filing date of '842 PCT application	6	458
April 28, 2010	Filing date of '835 patent's application	12	579
June 11, 2013	Filing date of '162 patent's application	20	1,353

28. Because the set of documents making up “the ADSL2 Standard G.992.3” shifts with time, a person having ordinary skill in the art would likely conclude that documents that did not exist on the filing date of the '842 PCT application were incorporated by reference in the '835 and '162 patents.

29. Therefore, in my opinion, both of the statements in Family 6 patents attempting to incorporate by reference various ADSL and VDSL standards did not identify with detailed particularity what specific material was being incorporated because the sets of documents they

identify changes as time passes. Furthermore, the ambiguities in the incorporation-by-reference statements could have been avoided: the drafter of the '842 PCT application presumably knew exactly which documents he wanted to incorporate by reference and could have set forth the titles of those specific documents and the dates on which they were made available to the public.

C. Technical Discussion

30. With a few exceptions such as those discussed below, and setting aside Dr. Chrissan's inappropriate reliance on the June 2004 version of G.993.1, I agree with much of Dr. Chrissan's explanation of data communication and of the specific interleaving and forward error correction techniques he chose to explain. Chrissan Dec. at ¶¶ 19-30. These techniques, and their benefits to mitigate the adverse effects of impulse noise, were well known at the time of the alleged invention and had been in use long before the priority date. For example, by 1993, forward error correction using Reed-Solomon coding in conjunction with convolutional interleaving had been incorporated into the draft of the original T1.413 standard for ADSL. *See, e.g.*, "Asymmetric Digital Subscriber Line (ADSL) WORKING DRAFT Standard" (T1E1.4/93-007) at §§ 6.2.1, 7.2.1 ("Up to four downstream simplex data channels and up to three duplex data channels are multiplexed and synchronized to the 4 kHz ADSL DMT symbol rate into two separate data buffers (fast and interleaved). A cyclic redundancy check (crc), scrambling, and forward error correction (FEC) coding are applied to each data buffer separately. The interleaved data buffer is then passed through an interleaving function."); 6.4.1 (describing use of Reed-Solomon coding); 6.4.2 ("The Reed-Solomon codewords in the interleave buffer are convolutionally interleaved."); 7.4.1 (describing use of Reed-Solomon coding). And, as indicated by Dr. Chrissan's heavy reliance on the contents of the 2002 version of G.992.3 to explain the technology, these techniques continued to be in common use as of the priority date of the Family 6 patents. *See, e.g.*, Chrissan Dec. at ¶¶ 19, 21-23, 26, 27, 31. Even the Family 6

patents themselves concede that “[i]t is standard practice for communications systems to use interleaving in combination with Forward Error Correction (FEC) to correct the errors caused by impulse noise.” ’835 patent at col. 1:34-37. Thus, to the extent Dr. Chrissan has presented data communication, error correction, and interleaving as if they were points of novelty of the alleged invention(s), I disagree with that presentation.

31. Furthermore, Dr. Chrissan’s explanation of the technology is incomplete in that he focuses entirely on one specific type of interleaving and one specific type of forward error correction, namely those used in ADSL and VDSL. But the Family 6 patents do not discuss or even mention the specific techniques of convolutional interleaving and Reed-Solomon forward error correction, and, as would have been appreciated by a person having ordinary skill in the art at the time of the alleged invention, other techniques for interleaving and forward error correction existed and were in use on the priority date of the ’842 PCT application. For example, other interleavers existing at the time of the alleged invention included block interleavers and pseudo-random interleavers. Although Dr. Chrissan explains that “[a] convolutional interleaver partitions its input into ‘blocks’ of bytes, and its operation is defined by two parameters: the block length ‘I’ and the depth parameter ‘D,’” (Chrissan Dec. at ¶ 23), he does identify any portion of the Family 6 patents that would lead a person having ordinary skill in the art to conclude that the interleaver presumed to exist in the context of the Family 6 patents is a convolutional interleaver. On the contrary, the Family 6 patents do not even mention “the block length ‘I’” that Dr. Chrissan contends is one of two parameters defining the operation of a convolutional interleaver. Furthermore, numerous forward error correction techniques other than Reed-Solomon coding existed at the time of the alleged invention, including, for example,

Hamming codes, turbo codes, Golay codes, low-density parity check (LDPC) codes, and many others.

32. I also disagree with certain assertions in Dr. Chrissan's explanation of the technology at issue. For example, I disagree with Dr. Chrissan's statement that "it would be incorrect to interpret the interleaver parameter D as being an interleaver depth 'in number of codewords.'" Chrissan Dec. at ¶ 32. In support of his statement, Dr. Chrissan offers only an example of his own making. Chrissan Dec. at ¶ 31. But Dr. Chrissan does not dispute the fact that the Family 6 patents themselves state that "D is the interleaver depth in number of codewords," ('835 patent at col. 2:13-14), nor does he explain why a skilled artisan would have ignored this part of the specification and instead made inferences based on the disclosures in a voluminous external document for a definition. Moreover, as explained above, the version of G.993.1 on which Dr. Chrissan and TQ Delta rely was not publicly available on the '842 PCT application's filing date. But even it states that the interleaver depth has units of number of codewords: "The interleave depth shall be programmable with a maximum interleave depth of 64 codewords." G.993.1 at § 8.4.1.

33. I agree with Dr. Chrissan that "[i]nterleaving (or an interleaver) takes an input byte stream (a byte is typically a group of eight bits) and then re-arranges and/or delays the input bytes according to a known pattern to produce an output byte stream," and that "[a]n interleaver is used in a transmitter, and a corresponding de-interleaver is used in a corresponding receiver to put the bytes back in the same order as the original input byte stream." Chrissan Dec. at ¶ 21.

VIII. FAMILY 6 PATENTS

34. I disagree with Dr. Chrissan's characterization of the Family 6 patents as providing "a novel solution for impulse noise adaptation." Chrissan Dec. at ¶ 33.

35. As he does in his explanation of the background technology, in his explanation of the Family 6 patents, Dr. Chrissan relies heavily on the disclosures of standards that, in my opinion, were inadequately identified in the '842 PCT application's incorporation-by-reference statements. *See, e.g.*, Chrissan Dec. at ¶¶ 34-35.

36. Much of the rest of Dr. Chrissan's overview of the Family 6 patents simply quotes the '835 patent. *See, e.g., id.* at ¶¶ 38-39.

IX. DISPUTED CLAIM TERMS

A. “transceiver”

37. In my opinion, a person having ordinary skill in the art at the time of the alleged invention would have understood “transceiver” to mean “communications device capable of transmitting and receiving data” as Defendants propose.

38. When looking at the Family 6 patents' specification, a person having ordinary skill in the art at the time of the alleged invention would not have understood that “the transmitter portion and receiver portion share at least some common circuitry.” The Family 6 patents do not include any written description or drawings to indicate that a transceiver includes common circuitry shared by its transmitter and receiver portions. In fact, there are no drawings illustrating a transceiver. Furthermore, the Family 6 patents do not include any explanation of what a transceiver is; instead, they describe only what a transceiver does. *See, e.g.*, '835 patent at col. 2:46-49 (transceivers enter into steady state data transmission); col. 4:32-36 (same); col. 4:52-54 (transceivers enter into Showtime); col. 5:7-10 (transceivers transmit and receive); col. 9:19-23 (purported advantage of disclosed technique is that transition between different FIP settings can be accomplished without reinitializing transceivers); col. 13:57-59 (“T” in “VTU-O” and “VTU-R” stands for “transceiver”); col. 13:66-67 (Showtime parameters exchanged between transceivers); col. 19:60-62 (message forwarded to transceiver).

39. The dictionary definitions Dr. Chrissan cites, all of which include the term “radio,” appear to be definitions specific to wireless transceivers. Chrissan Dec. at ¶ 40. In contrast, the Family 6 patents, and Dr. Chrissan himself, describe the alleged invention in the context of DSL. *See, e.g.*, ’835 patent at col. 11:66-15:60. Furthermore, other dictionary definitions from around the time of the alleged invention do not define a transceiver as including common circuitry. *See, e.g.*, A160 (Newton’s Telecom Dictionary, 2004), at 846; A163 (Oxford Dictionary of Computing, 2004), at 542.

40. Although the transmitter and receiver of a transceiver *can* share common circuitry, a person having ordinary skill in the art on the presumed priority date would not have understood shared common circuitry to be an inherent characteristic or a requirement of transceivers. Dr. Chrissan concedes as much when he states that “a digital signal processor (‘DSP’), a memory or storage component, a transmit/receive switch, an antenna and/or other components *could* be shared, *for example*, between the transmitter and receiver.” Chrissan Dec. at ¶ 41 (emphasis added).

41. Dr. Chrissan and TQ Delta contend that Defendants’ proposal “could incorrectly be interpreted to include those types of designs where the transmitter is isolated from, and functionally unrelated to, the receiver (e.g., communicating using an entirely different communication scheme over a different medium),” stating, without any support whatsoever, that such a scenario is “not contemplated by the patents-in-suit.” Chrissan Dec. at ¶ 43; TQ Delta Brief at 16. But neither TQ Delta nor Dr. Chrissan explains how TQ Delta’s proposal could not also “incorrectly be interpreted to include those types of designs where the transmitter is isolated from, and functionally unrelated to, the receiver (e.g., communicating using an entirely different communication scheme over a different medium).” Nothing in TQ Delta’s proposal forecloses

the possibility that the transmitter of a transceiver “is isolated from, and functionally unrelated to, the receiver (e.g., communicating using an entirely different communication scheme over a different medium).” For example, TQ Delta’s proposal does not exclude transceivers that transmit over one medium and receive over another medium, because, even in such a transceiver, the transmitter portion and receiver portion could share, for example, a processor.

42. TQ Delta appears to be concerned that “transceiver” could be construed to include systems in which the transmitter is in a separate physical enclosure from the receiver. As an initial matter, the Family 6 patents themselves state that “the components of the system can be arranged at any location within a distributed network without effecting [*sic*] the operation of the system,” (’835 patent at col. 7:50-52 (emphasis added)), which would lead a person having ordinary skill in the art to the conclusion that the transmitter and receiver need not be collocated. Thus, the patents themselves appear to contemplate the scenario TQ Delta fears. Moreover, TQ Delta’s concern is unfounded because Defendants’ proposed construction requires the transceiver to be a “communications device,” not “at least one communications device” or “one or more communications devices.” Furthermore, a skilled artisan would have understood that a “communications device” would not include systems that are entirely independent from one another.

43. In my opinion, TQ Delta’s proposed “common circuitry” requirement would introduce confusion into the term, because a person having ordinary skill in the art at the time of the alleged invention would not have known how much circuitry would have to be “common” in order to meet the claim. Consequently, in my opinion, based on the disclosures in the Family 6 patents, a person having ordinary skill in the art at the time of the alleged invention would have

understood a transceiver to be simply a “communications device capable of transmitting and receiving data” as Defendants have proposed.

B. “flag signal” (claims 8 and 10 of the ’835 patent)

44. In my opinion, based on the disclosures of the Family 6 patents, a person having ordinary skill in the art at the time of the alleged would have understood the “flag signal” in claims 8 and 10 of the ’835 patent to be a signal indicating when updated FIP settings are to be used.

45. The Family 6 patents state explicitly that the transmitter sends to the receiver a “Flag Signal indicating when updated FIP Parameters are to be used.” ’835 patent at FIG. 6. *See also id.* at col. 19:20-22 (“flag signal indicating when the new FIP settings are to be used”). Thus, Defendants’ construction is grounded directly in the written description and drawings.

46. Defendants’ proposal also comports with the meaning the term “flag signal” would have had to a person having ordinary skill in the art at the time of the alleged invention. The term “flag signal” was (and is) used in the electrical and computer engineering arts to refer to a signal that indicates something. *See, e.g.*, U.S. Patent No. 5,359,656 (issued Oct. 25, 1994) at col. 5:31-36 (“When both the absolute mean valued and the number of zero-crossing points *Z* exceed the first and the second thresholds TH1 and TH2, the flag signal of the decision circuits 32 and 38 becomes a logic ‘1’ level; and, otherwise, the flag signal of the decision circuits 32 and 38 becomes a logic ‘0’ level.”); *id.* at col. 5:55-68 (“As may be seen from Table 1, the adaptive selector 38 activates the switch 37 to select one of the three channels in accordance with the flag signals of the first and the second decision circuits 32 and 33. That is, when the flag signal of the first decision circuit 32 is of a logic ‘0’ level, the first channel is selected by the adaptive selector 38 and the switch 37, thereby coupling line 50 of the transceiver 40 to the ground. When the flag signal of the first decision circuit 32 is of a logic ‘1’ level and the flag

signal of the second decision circuit is logic ‘0’, the second channel is selected. When the flag signal of the first decision circuit 32 is logic ‘1’ and the flag signal of the second decision circuit 34 is logic 10, the third channel is selected.”); U.S. Patent No. 7,023,944 (filed Apr. 6, 2001) at col. 1:55-65 (“According to the conventional method, the clock PH_N or PH_N+1 is selected by determining whether the flag signal Flag_N or Flag_N+1 is enabled, e.g. at a high-level state). When the receiver detects that the phase of the data stream DATA has changed, the flag signal is changed correspondingly. For example, at time t, the flag signal Flag_N is changed from high to low while Flag_N+1 is changed from low to high.”); U.S. Patent Publication No. 2001/0000820 (published May 3, 2001) at ¶ [0010] (“The FLAG signal identifies the start of a command packet, and it also signals the start of an initialization sequence, as described in greater detail below.”); *id.* at ¶ [0029] (“As explained above, the initialization FLAG signal is a FLAG signal that is initially high for a duration that is twice the duration of the FLAG signal during normal operation. Thus, two adjacent FLAG bits, e.g.. F<O>and F<I>, can both be logic ‘1’. In contrast, as mentioned above, only one FLAG bit can be logic ‘1’ during normal operation of the memory device 16a. The NAND gate 212 (FIGS. 3 and 4) is used to detect the initialization FLAG signal by detecting when the F<O>and F<I> FLAG bits are both logic ‘1’.”) (emphases added).

47. TQ Delta’s proposed construction seeks to limit “flag signal” to two specific examples (or possibly only one example; see below) of “flag or marker signals” provided in the written description. In support of TQ Delta’s proposal, TQ Delta and Dr. Chrissan cite the portion of the Family 6 patents’ written description stating that the “flag or marker signal . . . is similar to that used in the ADSL2 G.992.3 ORL protocol.” TQ Delta Brief at 18; Chrissan Dec. at ¶ 47; ’835 patent at col. 12:1-5. Although the Family 6 patents do not identify any specific “flag or marker signal” from G.992.3 that is the referenced “that used in the ADSL2 G.992.3

ORL protocol,” TQ Delta and Dr. Chrissan conclude, without any further citation to the Family 6 patents, that “a flag or marker signal similar to that used in the ADSL2 G.992.3 OLR protocol is an inverted sync *signal*, also described as a sync flag.” TQ Delta Brief at 18; Chrissan Dec. at ¶ 47. I disagree with TQ Delta’s and Dr. Chrissan’s interpretation for several reasons.

48. First, the portion of the Family 6 patents cited by TQ Delta and Dr. Chrissan states only that the “flag or marker signal” is “*similar to* that used in the ADSL2 G.992.3 ORL protocol.” ’835 patent at col. 12:1-5 (emphasis added). A skilled artisan would not have understood “similar to” to mean “identical to” or “limited to.” Indeed, a skilled artisan would have considered such an interpretation to be in conflict with the portion of the Family 6 patents that states explicitly that the “inverted sync symbol” and “sync flag” to which TQ Delta now seeks to limit the definition of “flag signal” are merely exemplary: “*For example*, the flag signal *could be* an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol.” ’835 patent at col. 12:29-31 (emphasis added). Thus, a skilled artisan would not have understood “flag signal” to be limited only to an “inverted sync symbol” or “sync flag.”

49. Furthermore, at the time of the alleged invention, neither “inverted sync symbol” nor “sync flag” was a term of art, and neither term appears in the G.992.3 Recommendation in force on the priority date of the Family 6 patents. *See, e.g.*, G.992.3 (07/2002) (defining “sync symbol” but not “inverted sync symbol,” and not defining “sync flag”). The Family 6 patents themselves provide no definition for either “inverted sync symbol” or “sync flag,” and therefore a skilled artisan would not have known exactly what either an “inverted sync symbol” or “sync flag” is. Instead, he or she would have had to guess what might qualify as an “inverted sync symbol” or “sync flag.” Therefore, under TQ Delta’s proposed construction, a skilled artisan

would not have been able to determine with any certainty whether a design included the claimed “flag signal.”

50. Additionally, it is unclear from the Family 6 patents, TQ Delta’s opening brief, and Dr. Chrissan’s opening declaration whether there is even any difference between an “inverted sync symbol” and a “sync flag.” The punctuation used in the written description suggests that an “inverted sync symbol” might be the same thing as a “sync flag.” *See, e.g.*, ’835 patent at col. 12:29-31 (“the flag signal could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol.”). TQ Delta and Dr. Chrissan also seem to believe that an “inverted sync symbol” and “sync flag” are the same thing: “A flag or marker signal similar to that used in the ADSL2 OLR protocol is *an inverted sync signal, also described as a sync flag*.” TQ Delta Brief at 18; Chrissan Dec. at ¶ 47. Thus, TQ Delta’s proposed construction of “flag signal” seemingly includes redundant terms, neither of which was a term of art or had a well-understood meaning at the time of the alleged invention.

51. Because neither “inverted sync symbol” nor “sync flag” would have had a definite meaning to a skilled artisan, nor would a skilled artisan have known whether there was any difference between the two or what that difference might be, TQ Delta’s proposed definition of “flag signal” as “inverted sync symbol or sync flag used to synchronize the switch to using an updated FIP setting” would have been confusing and unclear to a skilled artisan at the time of the alleged invention. Given the difficulty a skilled artisan would face in interpreting TQ Delta’s proposed construction of “flag signal,” in my opinion, a lay jury would find it hopelessly confusing and unhelpful.

52. Moreover, although TQ Delta appears to attempt to limit the definition of “flag signal” to what it contends are flag signals used in ADSL2, the claims do not appear to be

limited to ADSL2. And, as explained above, a skilled artisan would have considered “flag signal” to encompass more than just ADSL2 signals. Even in the DSL field, flag signals were not unique to the ADSL2 OLR protocol. At the time of the alleged invention, a skilled artisan would have understood that flag signals were used in various other DSL systems and in the “handshake” protocol defined to allow DSL transceivers to initiate communication with each other. *See, e.g.*, G.991.1 (10/1998) at § I.5.5 (“Activation flags should be used to indicate pair(s) failure (e.g. LOS, LOST) to the mapping and maintenance functional block.”); § I.5.6 (“The transceiver which has flagged a failure to the mapping and maintenance functional block shall automatically initiate start-up procedures independently of the other transceivers in the system and continue to do so until it is able to reactivate the link. Once this has been achieved, the pair failure flag should be cleared.”). *See also* G.991.2 (12/2003) at § E.4.11.1.1 (“The PTM-TC frame format shall be as shown in Figure E.20. The opening and the closing Flag Sequences shall be set to 7E16. They identify the start and the end of the frame. Only one Flag Sequence is required between two consecutive frames.”); § E.4.11.2 (“To prevent failures due to false frame synchronization, any octet inside the PTM-TC frame that is equal to 7E16 (the Flag Sequence) or 7D16 (the Control Escape) shall be escaped as described below. After FCS computation, the transmitter examines the entire frame between the opening and the closing Flag Sequences. Any data octets which are equal to Flag Sequence or the Control Escape shall be replaced by a two-octet sequence consisting of the Control Escape octet followed by the original octet exclusive-OR’ed with 20₁₆. . . . On reception, prior to FCS computation, each Control Escape octet shall be removed and the following octet shall be exclusive OR’ed with 20₁₆ (unless the following octet is 7E16 which is the flag and indicates the end of the frame, and therefore an abort has occurred).”); § E.4.11.2.1 (flags used to detect invalid frames); § E.11.4.4 (“The PTM-TC frames

should be delineated by detecting of Flag Sequence. The incoming stream is examined on an octet-by-octet basis for the value 7E16. Two (or more) consecutive flag sequences constitute an empty frame (frames), which shall be discarded, and not counted as a FCS error.”); G.994.1 (05/2003) at § 3.4 (defining Galf as “[a]n octet of value 81₁₆, i.e., the ones complement of an HDLC flag.”); *id.* at § 8.2 (“Frames shall begin and end with standard HDLC flag octets (01111110) as defined in ISO/IEC 3309. At least three but not more than five flags shall be sent to begin a frame. At least two but not more than three flags shall follow the FCS of each frame.”); *id.* at § 11.1.1 (“When the HSTU-C has detected R-TONE1, it shall respond by transmitting Galfs on modulated carriers (C-GALF1). When the HSTU-R has detected Galfs, it shall respond by transmitting Flags on modulated carriers (R-FLAG1). When the HSTU-C has detected Flags, it shall respond by transmitting Flags (C-FLAG1). When the HSTU-R has detected Flags, it shall begin the first transaction.”); *id.* at § 11.3 (“After receiving the ACK(1) or NAK-CD message, the HSTU-R (HSTU-C) shall continue to transmit Flags for a period not to exceed 0.5 s. It shall then transmit 4 octets of Galf (referred to as R-GALF2 for an HSTU-R, C-GALF2 for an HSTU-C), followed by silence which terminates the G.994.1 session. When the HSTU-C (HSTU-R) detects either Galfs or silence, it shall continue to transmit Flags (referred to as C-FLAG2 for an HSTU-C, R-FLAG2 for an HSTU-R) for a period not to exceed 0.5 s, followed by silence which terminates the G.994.1 session.”).

53. Other of the inventor’s patent applications before the Family 6 patents’ parent application also support the conclusion that the “inverted sync symbol” and “sync flag” are merely two examples of “flag signals.” *See, e.g.*, U.S. Patent Pub. No. 2003/0112884 (filed Mar. 10, 2000) at ¶ [0034] (“In one embodiment, the synchronizing includes sending a flag signal. In another embodiment, the flag signal is a predefined signal. In a further embodiment, the

predefined signal is a sync symbol with a predefined phase shift. In a still further embodiment, the predefined signal is an inverted sync symbol.”) (emphasis added). Consequently, as the inventor himself understood, the term “flag signal” would have had a broader meaning to a skilled artisan than that proposed by TQ Delta.

54. TQ Delta and Dr. Chrissan suggest, incorrectly, that Defendants’ proposal requires the flag signal to “contain information ‘indicating when updated settings are to be used.’” TQ Delta Brief at 18; Chrissan Dec. at ¶ 48. As is clear from TQ Delta’s and Dr. Chrissan’s placement of quotation marks, Defendants’ proposal does not require the flag signal to *contain information* indicating when updated settings are to be used. It merely says, correctly, that the flag signal is a signal “indicating when updated FIP settings are to be used,” which comports with how a skilled artisan would have understood the term “flag signal” and tracks the disclosures in the written description and drawings. *See, e.g.*, ’835 patent at FIG. 6 (“Transmitter forward to Receiver Flag Signal indicating when updated FIP Parameters are to be used”); *id.* at col. 19:20-22 (“the transmitter forwards to the receiver a flag signal indicating when the new FIP settings are to be used.”); *id.* at col. 19:25-28 (“at the predefined change time following the reception of the flag signal, the receiver commences reception utilizing the new FIP parameters.”); *id.* col. 11:4-9 (“the receiver and transmitter can synchronize the modification of the FEC and interleaving parameters such that the both the transmitter and receiver start using the parameters at the same instant in time. This synchronization can be based on, for example, a synchronization using FEC codeword counters or a flag signal.”); *id.* at col. 12:8-11 (“For synchronization using a flag signal, the receiver and transmitter would start using updated FEC and interleaving parameters on a pre-defined FEC codeword boundary following the sync flag.”); *id.* at col. 12:25-37 (“The transmitting modem then sends a flag or marker signal to the

receiving modem 200 indicating that the new FIP settings are to be used on a predetermined number of DMT symbols following the transmission of the flag or marker signal. For example, the flag signal could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol. The transmitting modem 300 then starts using the new FIP settings for transmission on the predetermined number of DMT symbols following the transmission of the flag or marker signal. Similarly, the receiving modem starts using the new FIP settings for reception once the predetermined number of DMT symbols following the receipt of the flag or marker signal have been received.”).

55. TQ Delta and Dr. Chrissan argue that “the flag signal is used to synchronize the switch, but does not itself contain information ‘indicating when’ the switch will occur.” TQ Delta Brief at 19; Chrissan Dec. at ¶ 49. Although the written description does explain that “the receiver and transmitter can synchronize the modification of the FEC and interleaving parameters such that the both the transmitter and receiver start using the parameters at the same instant in time,” (’835 patent at col. 11:4-9), the asserted claims do not require any synchronization between the transmitter that transmits the flag signal and the receiver that receives the flag signal. The asserted claims claim a single apparatus comprising a single transceiver, and all of the claim limitations are directed to the transmitter portion of that transceiver. No part of the asserted claims is directed to any receiver on the other end of the communication channel that might receive and use the flag signal. *See, e.g.*, ’835 patent at col. 21:33-46 (“An apparatus . . . comprising: a transceiver, including a processor, configurable to: transmit a signal using a first FIP setting, transmit a flag signal, and switch to using for transmission, a second FIP setting following transmission of the flag signal, wherein . . . the switching occurs on a pre-defined forward error correction codeword boundary following the

flag signal.”). Thus, the asserted claims do not require any synchronization between the transmitter portion of the claimed transceiver, which transmits the flag signal, and the unclaimed receiver on the other end of the communication channel, which receives the flag signal. Therefore, in my opinion, a person having ordinary skill in the art at the time of the alleged invention would not have understood the “flag signal” to require or imply any action by the unclaimed receiver.

56. Furthermore, Defendants’ proposal does not exclude embodiments in which the flag signal is used in conjunction with “a pre-defined protocol known to both the transmitter and receiver.” TQ Delta Brief at 18; Chrissan Dec. at ¶ 49. Under Defendants’ construction, the flag signal indicates when updated FIP settings are to be used, which is not inconsistent with the transmitter and receiver also using a protocol with the flag signal.

57. The drafter of the Family 6 patents knew the terms “inverted sync symbol” and “sync flag” and could have chosen to use either term in the asserted claims. But he did not. Instead, he chose the term “flag signal,” which a skilled artisan would have understood to have a broader meaning based on the common usage of that term in the art and the statement in the Family 6 patents that “inverted sync symbol” and “sync flag” are just examples of a “flag signal.” ’835 patent at col. 12:29-30.

C. “flag signal” (claims 8 and 9 of the ’162 patent)

58. In my opinion, based on the disclosures of the Family 6 patents, a person having ordinary skill in the art at the time of the alleged invention would have understood the “flag signal” in claims 8 and 9 of the ’162 patent to be a signal indicating when an updated interleaver parameter setting is to be used. The Family 6 patents state explicitly that the transmitter sends to the receiver a “flag signal indicating when the new FIP settings are to be used.” ’835 patent at col. 19:20-22. *See also id.* at FIG. 6 (“Transmitter forward to Receiver Flag Signal indicating

when updated FIP Parameters are to be used.”). The acronym “FIP” stands for “forward error correction and interleaving parameters.” *See, e.g.*, ’835 patent at col. 2:23-24; 3:31; 8:9-10; claims 1, 8, 17, 24. In the context of claims 8 and 9 of the ’162 patent, the flag signal indicates when an updated interleaver parameter setting is to be used.

59. Like the Defendants’ construction of the term “flag signal” in the context of the ’835 patent, discussed above, ADTRAN’s construction is grounded directly in the written description and drawings and comports with how a person having ordinary skill in the art at the time of the alleged invention would have understood the term “flag signal” in the context of the ’162 patent.

60. In contrast, TQ Delta’s proposed construction would have been unclear and confusing to a skilled artisan because (a) neither “inverted sync symbol” nor “sync flag” is defined in the Family 6 patents or G.992.3, (b) “inverted sync symbol” and “sync flag” were not terms of art at the time of the alleged invention and would not have had a definite meaning to a skilled artisan, and (c) a skilled artisan would not have known whether there was any difference between an “inverted sync symbol” and a “sync flag” or what that difference might be. Given the difficulty a skilled artisan would face in interpreting TQ Delta’s proposed construction of “flag signal,” in my opinion, a lay jury would find it hopelessly confusing and unhelpful. In contrast, ADTRAN’S construction of “flag signal” as “signal indicating when an updated interleaver parameter value is to be used” would have been clear to a skilled artisan at the time of the alleged invention and would be clear to a lay jury now.

D. “interleaver parameter value”

61. In my opinion, in the context of the Family 6 patents, a person having ordinary skill in the art at the time of the alleged invention would have understood “interleaver parameter value” to mean the numerical value, in codewords, of the interleaver depth.

62. At the time of the alleged invention, “interleaver parameter value” was not a term of art and did not have a generally-understood meaning. Therefore, a skilled artisan would have looked to the Family 6 patents themselves to understand the meaning of this term.

63. The Family 6 patents do not use the term “interleaver parameter value” anywhere except in the claims. But a skilled artisan would have understood from the Family 6 patents that:

- The acronym “FIP” stands for “forward error correction (FEC) and interleaver parameters.” *See, e.g.*, ’835 patent at col. 2:23-24 (“FEC and Interleaving Parameters (FIP)”); *id.* at 3:31 (“For each FEC and Interleaving Parameter (FIP) setting . . .”); *id.* at 8:9-10 (“a forward error correction and interleaving parameter (FIP) module 230 . . .”); *id.* at claims 1, 8, 17, 24 (“forward error correction and interleaver parameter (FIP) settings”).
- The FIP parameters are N, K, R, and D. *See, e.g., id.* at col. 13:43-45 (“While these examples restrict the changes to a subset of the FIP parameters, they can obviously be extended to cover any combination of the FIP parameters (N, K, R and D).”).
- The FIP parameter “D” is the interleaver depth. *See, e.g., id.* at col. 5:57-59 (“Additionally, for example, the interleaving may be disabled by setting the interleaver depth to 1 (i.e., D=1).”); *id.* at col. 12:46-47 (“The Codeword Size (N) and Interleaver Depth (D) are not changed.”); *id.* at col. 12:62-64 (“The FEC Codeword Size (N=K+R) and Interleaver Depth (D) are not changed.”); *id.* at col. 13:7-13 (“This is the case because if the codeword size N and the interleaver depth D are not modified. . . . [M]emory locations are overwritten due to on-line changes in the codeword size (N) and or interleaver depth (D).”); *id.* at col. 13:23-25 (“The number of information bytes in a codeword (K) and Interleaver Depth (D) are not changed. . .

.”); *id.* at col. 13:49-53 (“In order to keep the memory and latency constant it is necessary to change the codeword size (N) accordingly when changing the interleaver depth (D). For example, if the interleaver depths [*sic*] is changed from D=64 to D=128, the Codeword size would have to be decreased by a factor of 2 so that overall latency is constant.”).

- D is a value. *See, e.g.*, col. 13:43-47 (“While these examples restrict the changes to a subset of the FIP parameters, they can obviously be extended to cover any combination of the FIP parameters (N, K, R and D). For example, the value of D could also be modified in addition to the values of K, R and N.”).
- D is the interleaver depth in number of codewords. *See, e.g.*, col. 2:13-14 (“D is the interleaver depth in number of codewords”).

From these disclosures, and based on the fact that the Family 6 patents do not mention any parameter or setting for the interleaver *other* than the interleaver depth D, which is given in units of codewords, a person having ordinary skill in the art at the time of the alleged invention would have understood “interleaver parameter value” to mean “the numerical value of the interleaver depth in number of codewords” as Defendants have proposed.

64. Dr. Chrissan and TQ Delta contend that the interleaver depth should not be expressed in codewords. In attempting to support their position, Dr. Chrissan and TQ Delta ignore the disclosures of the Family 6 patents entirely and focus solely on what a skilled artisan could potentially infer from a version of G.993.1 that, as explained above, would not have been available to a person having ordinary skill in the art at the time of the alleged invention. TQ Delta Brief at 20; Chrissan Dec. at ¶ 53. Therefore, at the time the ’842 PCT application was filed, a skilled artisan would not have considered the disclosures of G.993.1 to determine the

meaning of “interleaver parameter value.” Furthermore, as noted above, the version of G.993.1 on which TQ Delta and Dr. Chrissan improperly rely itself states that the interleaver depth has units of codewords. *See* G.993.1 at § 8.4.1 (“The interleave depth shall be programmable with a maximum interleave depth of 64 codewords.”).

65. Despite the fact that the Family 6 patents state that “D is the interleaver depth in number of codewords,” (’835 patent at col. 2:13-14), and the Family 6 patents disclose no alternative units for the interleaver depth, Dr. Chrissan and TQ Delta appear to take the extraordinary position that “interleaver parameter value” is either a unitless quantity or can have any units at all (except, that is, codewords). *See, e.g.*, TQ Delta Brief at 20; Chrissan Dec. at ¶ 53 (“the manner in which the interleaver parameter value is used in each of the example DSL standards that are incorporated by reference in the Family 6 patents does not require the interleaver depth to be expressed in any particular units, much less in codewords.”). In my opinion, without *some* specified units, the value of the interleaver depth, and, therefore, the “interleaver parameter value,” would have been meaningless to a person having ordinary skill in the art at the time of the alleged invention. Specifically, absent knowledge of the units of “interleaver parameter value,” a skilled artisan would not have understood how to avoid infringing the claims, because “interleaver parameter value” could be any value at all in any conceivable units. The only units for the interleaver depth specified in the Family 6 patents are codewords, and, consequently, those are the units a skilled artisan would have understood the interleaver depth to have.

66. Moreover, TQ Delta argues that “interleaver depth is simply a number that defines, per the DSL standards, how the byte position of bytes of the input blocks will be delayed (and therefore, spaced apart) in the interleaved output.” TQ Delta Brief at 21. As an initial

matter, a person having ordinary skill in the art at the time of the alleged invention would not have understood the term “interleaver depth” to be “a number that defines, per the DSL standards, how the byte position of bytes of the input blocks will be delayed in the interleaver output.” A skilled artisan would not have understood the term “interleaver depth” to refer to any specific type of interleaver, much less the convolutional interleaver used in DSL but never discussed or even mentioned in the Family 6 patents. Moreover, TQ Delta does not appear to contend that the asserted claims are limited to DSL, and the claims themselves do not appear to be so limited. Therefore, in my opinion, TQ Delta’s reliance on “the DSL standards” to define “interleaver depth” is inappropriate.

67. A skilled artisan would have understood that the interleaver depth is one plus the minimum separation, in bytes, at the output of the interleaver between any two bytes that were adjacent at the input of the interleaver. *See, e.g.,* Cioffi Course Reader, Chapter 11 at 258 (available at <https://web.stanford.edu/group/cioffi/ee379b/reader.html>).

68. Defendants’ proposed construction is commensurate with both the Family 6 patents’ specification and how a skilled artisan would have understood “interleaver parameter value” on the filing date of the Family 6 patents’ ultimate parent application.

E. “FIP setting”

69. In my opinion, a person having ordinary skill in the art at the time of the alleged invention would have understood “FIP setting” to mean “forward error correction and interleaver parameters characterized by the set of parameters for codeword size in bytes, number of information bytes in a codeword, number of parity or redundancy bytes in a codeword, and interleaver depth in codewords” as Defendants have proposed.

70. Dr. Chrissan asserts, without any support whatsoever, that “[a]n ‘FIP setting,’ or ‘FEC and Interleaving Parameters setting,’ is a set including at least one forward error correction

parameter value and at least one interleaver parameter.” Chrissan Dec. at ¶ 54. Dr. Chrissan then argues again—using only the June 2004 G.993.1 Recommendation that, as explained previously, was not publicly available to a skilled artisan on the filing date of the ’842 PCT application—that the interleaver depth is not necessarily in codewords. *Id.* at ¶ 55. For the reasons set forth below, I disagree.

71. At the time of the alleged invention, neither the acronym “FIP” nor the term “FIP setting” was a term of art, and neither had a generally-understood meaning. Therefore, contrary to TQ Delta’s assertion, at the time of the alleged invention, there was no “understanding of the term by a person of ordinary skill in the art.” TQ Delta Brief at 21. For example, prior to reading the Family 6 patents, I do not recall ever having seen the acronym “FIP.” In my opinion, the acronym “FIP” is peculiar to the Family 6 patents and, consequently, a skilled artisan would have looked only to the Family 6 patents to understand the meaning of “FIP” and “FIP setting.”

72. As explained above, the Family 6 patents state that the acronym “FIP” stands for “forward error correction and interleaver parameters.” *See, e.g.*, ’835 patent at col. 2:23-24 (“FEC and Interleaving Parameters (FIP)”); *id.* at 3:31 (“For each FEC and Interleaving Parameter (FIP) setting . . .”); *id.* at 8:9-10 (“a forward error correction and interleaving parameter (FIP) module 230 . . .”); *id.* at claims 1, 8, 17, 24 (“forward error correction and interleaver parameter (FIP) settings”).

73. The Family 6 patents also indicate that the FIP parameters are all of N, K, R, and D. *See, e.g., id.* at col. 13:43-45 (“While these examples restrict the changes to a subset of the FIP parameters, they can obviously be extended to cover any combination of the FIP parameters (N, K, R and D).”).

74. The Family 6 patents also indicate that the FIP settings are the values of N, K, R, and D. *See, e.g., id.* at col. 3:37-47 (“A plurality of FIP settings can be used for transmission and reception. In accordance with one particular aspect of this invention, the system can transition from one FIP setting to another FIP setting without going through the startup initialization procedure such as the startup initialization sequence utilized in traditional xDSL systems. For example, an xDSL system that implements the systems and methods described herein could start using an FIP setting of (N=255, K=247, R=S, D=64) and then transition to an FIP setting of (N=255, K=239, R=16, D=64) without re-executing the startup initialization procedure.”).

75. The Family 6 patents also disclose that K is the number of information bytes in a codeword (*see, e.g., id.* at col. 2:16 (“K is the number of information bytes in a codeword”)); R is the number of parity or redundancy bytes in a codeword (*see, e.g., id.* at col. 2:12-13 (“R is the number of parity (or redundancy) bytes in a codeword”)); and N is the FEC codeword size in bytes (*see, e.g., id.* at col. 2:12 (“N is the codeword size in bytes”)), which is equal to K+R (*see, e.g., id.* at col. 2:18 (“N=K+R”)). And, as explained above, the Family 6 patents state that D is the interleaver depth in number of codewords (*see, e.g., col. 2:13-14* (“D is the interleaver depth in number of codewords”))).

76. TQ Delta argues, without any support from Dr. Chrissan or the Family 6 patents, that “Defendants’ proposed construction improperly limits the claim term to require all of a set of examples disclosed in the specification.” TQ Delta Brief at 21. But, as explained above, the term “FIP setting” was not a term of art and would not have had any meaning to a skilled artisan at the time of the alleged invention, so a skilled artisan would have given great weight to the examples provided in the Family 6 patents. And the term “FIP setting” as used in the Family 6 patents *always* refers to all four of the FIP parameters included in Defendants’ construction, even

if, in particular examples, fewer than all of the FIP parameters are modified. *See, e.g.*, '835 patent at 3:38-47 ("In accordance with one particular aspect of this invention, the system can transition from one FIP setting to another FIP setting without going through the startup initialization procedure such as the startup initialization sequence utilized in traditional xDSL systems. For example, an xDSL system that implements the systems and methods described herein could start using an FIP setting of (N=255, K=247, R=S, D=64) and then transition to an FIP setting of (N=255, K=239, R=16, D=64) without re-executing the startup initialization procedure."); *id.* at col. 12:43-50 ("This section describes an example of FIP settings for On-Line INP adaptation for DSL. In this example only the number of information bytes in a codeword (K) and the number of parity bytes in a codeword (R) are updated on-line. The Codeword Size (N) and Interleaver Depth (D) are not changed. This means that the latency (and interleaver memory size) and the line rate are not modified on-line. Since $N=K+R$ this places restrictions on the allowed values for K and R."); *id.* at col. 13:4-10 ("This means provided that the modification if [sic] the FIP setting is restricted to K and R, the transition between FIP settings can be done in a seamless manner. This is the case because if the codeword size N and the interleaver depth D are not modified, the transition can happen without the problem of 'interleaving memory flushing.'"); *id.* at col. 13:20-28 ("This section describes an example of FIP settings for On-Line INP adaptation for DSL. In this example only the Codeword Size (N) and the number of parity bytes in a codeword (R) are updated on line. The number of information bytes in a codeword (K) and Interleaver Depth (D) are not changed and therefore the user data rate does not change. This means that the latency (and interleaver memory size) and the line rate are modified on-line. Since $N=K+R$ this places restrictions on the allowed values for K and R.") (emphases added).

77. TQ Delta also argues that the construction of “FIP setting” should not specify units for the FIP parameters N, K, R, and D, and that the “plain meaning of the ‘FIP setting’ term” should control. TQ Delta Brief at 22. But, as explained above, there was (and is) no “plain meaning” of “FIP setting.” Although the FIP settings are described in the context of DSL, there is no indication in the Family 6 patents that the units of these settings’ values (*i.e.*, N, K, and R in bytes, and D in codewords) would be different for other kinds of systems. For example, the Family 6 patents do not provide any non-DSL examples of FIP settings. Moreover, without *some* units for each of the FIP parameters, the parameters N, K, R, and D would have been meaningless to a person having ordinary skill in the art at the time of the alleged invention. The only units specified for N, K, R, and D in the Family 6 patents are those included in Defendants’ construction.

78. Because the term “FIP settings” was not a term of art at the time of the alleged invention and appears to be a term peculiar to the Family 6 patents, a person having ordinary skill in the art would have understood “FIP settings” to be “forward error correction and interleaver parameters characterized by the set of parameters for codeword size in bytes, number of information bytes in a codeword, number of parity or redundancy bytes in a codeword, and interleaver depth in codewords” as Defendants have proposed.

F. “FIP value”

79. In my opinion, a skilled artisan at the time of the alleged invention would have understood the term “FIP value” to mean “numerical value of codeword size in bytes, number of information bytes in a codeword, number of parity or redundancy bytes in a codeword, or interleaver depth in number of codewords” as Defendants have proposed.

80. Despite the fact that the Family 6 patents state that “D is the interleaver depth in number of codewords,” (’835 patent at col. 2:13-14), TQ Delta and Dr. Chrissan argue that

“Defendants’ use of the phrase ‘interleaver depth in number of codewords’ in their construction is incorrect.” TQ Delta Brief at 23; Chrissan Dec. at ¶ 57. Once again, TQ Delta and Dr. Chrissan focus entirely on the disclosures of a version of G.993.1 that would not have been available to, and would not have been considered by, a skilled artisan trying to determine the meaning of “FIP value” at the time of the alleged invention. TQ Delta Brief at 20; Chrissan Dec. at ¶ 53. But even that document states that the interleaver depth has units of codewords. *See* G.993.1 at § 8.4.1 (“The interleave depth shall be programmable with a maximum interleave depth of 64 codewords.”). The Family 6 patents do not disclose or discuss any units for the interleaver depth *except* codewords. Thus, a person having ordinary skill in the art at the time of the alleged invention would have understood the interleaver depth to have units of codewords in the context of the Family 6 patents.

81. TQ Delta argues that its proposed construction “relies on a plain meaning of the claim term.” TQ Delta Brief at 23. But, as explained above, “FIP” was not a term of art and did not have a generally-understood meaning at the time of the alleged invention. Therefore, there was (and is) no plain meaning for the term “FIP” or “FIP value,” and a skilled artisan at the time of the alleged invention would have looked to the Family 6 patents’ disclosures to understand what a “FIP value” is.

82. The Family 6 patents do not define the term “FIP value” or provide examples of “FIP values,” but a person having ordinary skill in the art would recognize based on the disclosures of the Family 6 patents that a “FIP value” is simply the value of one of the “FIP parameters.” As explained above, the “FIP parameters” are N, K, R, and D. *See, e.g.*, ’835 patent at col. 13:43-45 (“While these examples restrict the changes to a subset of the FIP parameters, they can obviously be extended to cover any combination of the FIP parameters (N,

K, R and D).”). Thus, a skilled artisan would have understood “FIP value” to be the “numerical value of codeword size in bytes, number of information bytes in a codeword, number of parity or redundancy bytes in a codeword, or interleaver depth in number of codewords.”

G. “the switching occurs on a pre-defined forward error correction codeword boundary”

83. In my opinion, a person having ordinary skill in the art at the time of the alleged invention would not have known how to interpret the term “the switching occurs on a pre-defined forward error correction codeword boundary” because he or she would not have known, with a reasonable degree of certainty, what the “pre-defined forward error correction codeword boundary” would be or how to avoid infringing the asserted claims.

84. Dr. Chrissan and TQ Delta argue that because transceivers implementing “the incorporated DSL standards” “know” the positions of the FEC codeword boundaries, the term “the switching occurs on a pre-defined forward error correction codeword boundary” is definite. TQ Delta Brief at 24; Chrissan Dec. at ¶ 59. But simply knowing the positions of the FEC codeword boundaries is not enough because the asserted claims recite that the switch to the second FIP setting occurs “on a *pre-defined* forward error correction codeword boundary.” Thus, it is necessary to know not only *where* the FEC codeword boundaries are, but also *which* of those boundaries is the claimed “pre-defined” one. Just knowing where the FEC codeword boundaries are would not enable a person having ordinary skill in the art to determine where the “pre-defined forward error correction codeword boundary” is.

85. As explained above, a person having ordinary skill in the art would not understand exactly which standards, if any, had been incorporated by reference in the Family 6 patents because the set of documents identified by “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” and “the ADSL2 Standard G.992.3” changes

with time. In addition, even if some DSL standards documents were incorporated by reference on the '842 PCT application's filing date, those documents did not define any "pre-defined forward error correction codeword boundary" where a transceiver would "switch to using for transmission a second FIP setting" as required by the asserted claims of the '835 patent. Thus, even if a skilled artisan had been able to identify incorporated documents, he or she would not have found in those documents any definition of or way to determine the claimed "pre-defined forward error correction codeword boundary."

86. Furthermore, the asserted claims do not appear to require the use of a standard, and TQ Delta and Dr. Chrissan do not contend that a standard is necessary to practice the claims. *See, e.g.*, TQ Delta Brief at 24; Chrissan Dec. at ¶ 60 ("pre-defined" means that both the transceiver and receiver independently know how to calculate an exact FEC codeword boundary transition point relative to an occurrence of the sync flag, based on a protocol defined by a communication standard document *and/or by the designer of the transceiver devices.*") (emphasis added). In fact, as explained above, the asserted claims do not require any cooperation from or coordination with a receiver, nor do they require a receiver to respond in any way or perform any action. Therefore, the claims do not require any protocol at all, much less a "protocol defined by a communication standard document and/or by the designer of the transceiver devices." TQ Delta Brief at 24; Chrissan Dec. at ¶ 60.

87. The Family 6 patents describe switching to an updated FIP setting on an FEC codeword boundary *only* in the context of what the patents refer to as "synchronization using FEC codeword counters," ('835 patent at col. 11:7-9), which is different from synchronization using flag signals. Specifically, the written description states that "[f]or synchronization using FEC codeword counters, the receiver 200 and transmitter 300 can synchronize the change in

cooperation with the sync modules 280 and 320, by *counting the FEC codewords* from the beginning of Showtime and *the transition would occur when a specific FEC codeword counter value that is known by both the transmitter 300 and the receiver 200 is reached.*” ’835 patent at 11:10-16. *See also id.* at col. 11:47-57 (“The synchronization module 280 and message module 290 then cooperate to send a message to the transmitting modem 300 specifying the FEC codeword counter value on which the new FIP settings are to be used for transmission and reception. Alternatively, the transmitting modem, in cooperation with the message module 330, can send a message to the receiving modem indicating the FEC codeword counter value on which the FIP settings are to be used for transmission and reception. For example, the message can indicate that when the code word counter equals 501, the new FIP setting will be used for transmission and reception.”); col. 11:58-65 (“When the transmitter FEC codeword counter equals the value indicated in the message, the synchronization module 220 instructs the transmitting modem 300 to transition to the new FIP settings. Similarly, when the synchronization module 280 in the receiving modem 200 counts the FEC codeword that equals the value indicated in the message, the receiving modem 200 transitions to using the new FIP settings for reception.”).

88. The Family 6 patents make clear that embodiments using FEC codeword counters are different from embodiments using flag signals. For example, after explaining “synchronization using FEC codeword counters,” the written description continues by stating that “[s]ynchronization can also be performed through the use of a flag signal” to “synchronize the change in FIP settings using a flag or marker signal.” *Id.* at col. 11:66-12:4. The Family 6 patents go on to say that “[t]his protocol *may be more desirable than using an FEC codeword counter* because, for example, it has greater impulse noise immunity.” *Id.* at col. 12:5-7.

Although the written description then states that “[f]or synchronization using a flag signal, the receiver and transmitter would start using updated FEC and interleaver parameters on a pre-defined FEC codeword boundary following the sync flag,” (’835 patent at col. 12:8-11), the explanation that follows states repeatedly that the transmitter switches to using updated FIP settings not on a pre-defined FEC codeword boundary following the flag signal, but rather “on a *predetermined number of DMT symbols* following the transmission of the flag signal.” *See, e.g.*, ’835 patent at col. 12:25-37 (“The transmitting modem then sends a flag or marker signal to the receiving modem 200 indicating that the new FIP settings are to be used on a predetermined number of DMT symbols following the transmission of the flag or marker signal. . . . The transmitting modem 300 then starts using the new FIP settings for transmission on the predetermined number of DMT symbols following the transmission of the flag or marker signal. . . .”).

89. As would have been understood by a person having ordinary skill in the art at the time of the alleged invention, a DMT symbol is not the same as an FEC codeword boundary. The Family 6 patents do not describe any relationship between FEC codeword boundaries and DMT symbol boundaries. Therefore, a skilled artisan would have recognized that an FEC codeword can span multiple DMT symbols, and a single DMT symbol can contain bytes from multiple FEC codewords, and, as a result, FEC codeword boundaries do not necessarily align with DMT symbol boundaries. Consequently, a skilled artisan would understand that a switch occurring on a predetermined DMT symbol boundary, as described in the written description, does not imply that the switch also occurs on an FEC codeword boundary, as recited in the claims.

90. Thus, although the written description discloses—exactly once in the context of embodiments using a flag signal—that a pre-defined FEC codeword boundary exists and that “the receiver and transmitter would start using updated FEC and interleaving parameters on a pre-defined FEC codeword boundary following the sync flag,” (’835 patent at col. 12:8-11), it does not say anything about where the pre-defined FEC codeword boundary comes from (*e.g.*, who defines it, where or how it is defined, etc.) or what its value might be. Consequently, a skilled artisan would have concluded that the specification discloses the desired result—that the transmitter switches to using the updated FIP parameters on a pre-defined FEC codeword boundary following the flag signal—but does not disclose how to achieve that result. For example, the Family 6 patents do not disclose on which codeword the switch takes place, nor do they discuss whether or how FEC codewords should be counted when a flag signal is used. In fact, a skilled artisan would conclude that although FEC codeword counters are used “[f]or synchronization using FEC codeword counters,” they are *not* used with flag signals. *See, e.g.*, ’835 patent at col. 11:10-65 (discussing FEC codeword counter values in the context of synchronization using FEC codeword counters); *id.* at col. 11:66-12:37 (discussing synchronization using a flag signal with no mention of FEC codeword counters and stating protocol using flag signal “may be more desirable than using an FEC codeword counter because, for example, it has greater impulse noise immunity”). Therefore, a skilled artisan would have understood that the disclosures related to “synchronization using FEC codeword counters” would not apply to embodiments using a flag signal.

91. In my opinion, a skilled artisan would not have been able to identify the “pre-defined forward error correction codeword boundary” with any reasonable certainty, and

therefore he or she would not have understood the term “the switching occurs on a pre-defined forward error correction codeword boundary” or how to avoid infringing the asserted claims.

H. “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary”

92. In my opinion, a person having ordinary skill in the art at the time of the alleged invention would not have known how to interpret the term “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary” because he or she would not have known with reasonable certainty what the “pre-defined forward error correction codeword boundary” would be.

93. Dr. Chrissan and TQ Delta argue that “the same protocol is designed into the transmitter and the receiver, and both know the FEC codeword boundaries.” TQ Delta Brief at 25; Chrissan Dec. at ¶ 63. But, as explained above, just knowing the positions of the FEC codeword boundaries is not enough because the asserted claims recite that the switch to the second interleaver parameter value occurs “on a *pre-defined* forward error correction codeword boundary.” Thus, it is necessary to know not only *where* the FEC codeword boundaries are, but also *which* of those boundaries is the claimed “pre-defined” one. Just knowing where the FEC codeword boundaries are would not enable a person having ordinary skill in the art to determine where the “pre-defined forward error correction codeword boundary” is.

94. Furthermore, the asserted claims of the ’162 patent do not require any cooperation from or coordination with a receiver, nor do they require a receiver to respond in any way or perform any action. Therefore, the claims do not require any protocol at all, much less a “the same protocol” in the transmitter and receiver. TQ Delta Brief at 25; Chrissan Dec. at ¶ 63.

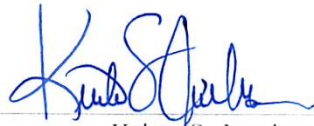
95. In addition, as discussed above, a skilled artisan would have concluded that the specification discloses the desired result—that the transmitter switches to using the updated

interleaver parameter value on a pre-defined FEC codeword boundary following the flag signal—but not how to achieve that result for embodiments using flag signals. Specifically, at the time of the alleged invention, a person having ordinary skill in the art would not have been able to determine, for flag signal embodiments, where the pre-defined FEC codeword boundary comes from or what its value might be or how to avoid infringing the asserted claims of the '162 patent.

96. In my opinion, a skilled artisan would not have been able to identify the “pre-defined forward error correction codeword boundary” with any reasonable degree of certainty, and therefore he or she would not have understood the term “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary.”

I declare under penalty of perjury of the laws of the United States that the foregoing is true and correct.

Date: October 18, 2017

A handwritten signature in blue ink, appearing to read 'Krista S. Jacobsen', is written over a horizontal line.

Krista S. Jacobsen

Krista S. Jacobsen

1999 S. Bascom Ave., Suite 700, Campbell, CA 95008
E-mail: krista@jacobseniplaw.com; Tel: 408.455.5539 (M)

Experience

JACOBSEN IP LAW, Campbell, CA (February 2014 - present)

Attorney and Counselor at Law. Solo practitioner providing expert consultant, expert witness, patent litigation support, patent prosecution, and intellectual property (IP) counseling services.

SANTA CLARA UNIVERSITY SCHOOL OF LAW, Santa Clara, CA (January 2015 - present)

Lecturer in Law. Co-teaching Pretrial Litigation Techniques (Fall 2015, Fall 2016, Fall 2017) and Law Practice Management (Spring 2015, Spring 2016, Spring 2017).

DISRUPTIVE FORCE LLC, Campbell, CA (February 2015 - present)

Co-founder and CEO.

HEADWATER PARTNERS, Redwood Shores, CA (July 2011 - February 2014)

Head Counsel. Responsibilities included general counsel duties and patent prosecution.

COVINGTON & BURLING, LLP, Redwood Shores, CA (October 2009 - July 2011)

Associate. IP litigation.

HELLER EHRMAN, Menlo Park, CA (May-July 2008)

Summer Associate. IP litigation. Researched and drafted legal memoranda and briefs.

BLAKELY SOKOLOFF TAYLOR & ZAFMAN, Sunnyvale, CA (June-August 2007)

Summer Associate. Patent prosecution. Composed responses to United States Patent and Trademark Office office actions and wrote portions of patent applications, including claims.

CONSULTANT (July 2004 - September 2007)

Responsibilities included assisting clients to determine and execute digital subscriber line (DSL) standardization and product strategies, writing simulations, generating and presenting technical tutorials, drafting and prosecuting patent applications, and helping with other patent issues. Clients included 2Wire, Inc., PMC-Sierra, Atheros Communications, and Beyer Law Group.

TEXAS INSTRUMENTS (TI) (formerly AMATI COMMUNICATIONS), San Jose, CA (January 1994 - May 2004).

Lead xDSL Standards Strategist (2001-04)

- Led TI's DSL standardization efforts, managed standards resources and budget, and communicated standards progress and status within TI.
- Developed and presented dozens of technical proposals for regional and international standards organizations, including ETSI TM6, ITU-T SG15/Q4, T1E1.4, and IEEE 802.3ah (Ethernet in the first mile).
- Wrote DSL white papers, technical reports, technical book chapters, and articles for external publications.
- Participated as a member of business unit patent committee and worked with TI legal department to ensure protection of intellectual property and to provide technical assistance.
- Wrote and ran DSL system simulations.
- Made presentations at industry conferences and events.

Previous roles with Amati/TI were *Project Manager, ADSL over ISDN* (2000-01), *Project Manager, VDSL* (1999-2000), *Senior Staff Engineer* (1998-99), *Staff Engineer* (1997-98), *Senior Systems Engineer* (1996-97), *Consultant* (1994-96).

VITEL COMMUNICATIONS CORPORATION, Santa Clara, CA (August 1991 - May 1992).

Member of telecommunications group. Worked on a team that developed a video telephony transceiver.

Bar Memberships and Registrations

STATE BAR OF CALIFORNIA

Member No. 267868 (admitted December 7, 2009)

COLORADO STATE BAR

Registration No. 43,294 (admitted May 24, 2011 (currently inactive))

UNITED STATES PATENT AND TRADEMARK OFFICE

Registration No. 59,374 (registered October 11, 2006)

Krista S. Jacobsen

Education

SANTA CLARA UNIVERSITY SCHOOL OF LAW (Santa Clara, CA)

JD (*magna cum laude*), 2009

Honors and Awards

Santa Clara University School of Law Intellectual Property (IP) Fellowship (2006-09)

Dean's List (2006-09)

Order of the Coif (2009)

ABA-BNA Award for Excellence In the Study of Intellectual Property Law (Spring 2008)

CALI Award for Excellence In: Managing Complex IP Litigation (Fall 2008), IP Litigation Techniques (Spring 2008), Patent Prosecution (Spring 2008), Protection of IP (Spring 2008), Advocacy (Fall 2007), Contracts (Spring 2007)

Witkin Award for Excellence In: Opening Statements and Closing Arguments (Fall 2008), Pretrial Litigation Techniques (Fall 2008), Mass Communication (Fall 2007), Property (Spring 2007), Legal Analysis, Research and Writing, Section H (Spring 2007)

Grand Prize, First Annual San Francisco Intellectual Property Law Association Student Writing Competition (February 2009)

Selected to represent Santa Clara University in the Dean Jerome Prince Memorial Evidence Moot Court Competition (Spring 2009) and Constance Baker Motley National Moot Court Competition in Constitutional Law (Spring 2008)

Activities

Co-President, Santa Clara University Student Intellectual Property Law Association (2008-09)

Vice President, Alumni Relations, Santa Clara University Intellectual Property Association (2007-08)

Associate, Santa Clara Computer and High Tech Law Journal (2007-08)

STANFORD UNIVERSITY (Stanford, CA)

Ph.D. in electrical engineering, 1996

Dissertation: *Discrete Multi-Tone-Based Communications in the Reverse Channel of Hybrid Fiber-Coax Networks*

Adviser: John M. Cioffi (<http://www.stanford.edu/group/cioffi>)

MSEE, 1993

Digital communications specialization

Honors and Awards

IBM Graduate Fellowship (1994-95)

National Science Foundation Graduate Fellowship (1991-94)

IEEE Communications Society Scholarship (1993)

UNIVERSITY OF DENVER (Denver, CO)

BSEE (*summa cum laude*), 1991

Communications specialization

Honors and Awards

Winner, Denver Section IEEE Student Paper Contest (1991)

University of Denver Pioneer Award (1991)

University of Denver Distinguished Senior Woman Award (1990)

Phi Beta Kappa (1988)

University of Denver Honors Scholarship (1986-91)

Colorado Scholars Scholarship (1987-91)

Publications

LAW REVIEW AND JOURNAL PAPERS

Krista S. Jacobsen, *Intellectual Property in Standards: Does Antitrust Law Impose a Duty to Disclose (Even If the Standards-Setting Organization Does Not)?*, 26 Santa Clara Computer & High Tech. L.J. 459 (2010).

Krista S. Jacobsen, *Methods, Marking, and Messiness: Revisiting the Federal Circuit's Rule That Product Marking is not Required Where a Patent is Directed to a Method*, 13 U.S.F. Intell. Prop. L. Bull. 107 (2009).

Krista S. Jacobsen

TECHNICAL BOOKS

Implementation and Applications of DSL Technology (Philip Golden, Herve Dedieu, and Krista S. Jacobsen, eds., Auerbach Publications, 2008): Co-author of chapter 17, entitled “DSL Standardization,” and author of the ADSL portion of that chapter.

Fundamentals of DSL Technology (Philip Golden, Herve Dedieu, and Krista S. Jacobsen, eds., Auerbach Publications, 2006): Author of chapter 7, entitled “Fundamentals of Multi-carrier Modulation.”

Author of the chapter entitled “Last Mile Copper Access” in *Broadband Last Mile Access Technologies for Multimedia Communications* (Nikil Jayant, ed., CRC Press, 2005).

Author of the definition of “broadband communication” in *World Book Encyclopedia* (2003).

TECHNICAL JOURNAL PAPERS AND MAGAZINE ARTICLES

K.S. Jacobsen. “Patents and Standardization, Part 3: Commitments to License Standard-Essential Patents Under Reasonable and Non-Discriminatory (RAND) Terms.” *IEEE Communications Magazine - Communications Standards Supplement*, September 2016, pp. 66-71.

K.S. Jacobsen. “Patents and Standardization, Part 2: Duties to Disclose Patents to Standards Development Organizations.” *IEEE Communications Magazine - Communications Standards Supplement*, July 2016, pp. 18-23.

K.S. Jacobsen. “Patents and Standardization, Part 1: A Tutorial on Patents.” *IEEE Communications Magazine - Communications Standards Supplement*, March 2016, pp. 10-14.

B. Wiese and K.S. Jacobsen. “Use of the Reference Noise Method Bounds the Performance Loss Due to Upstream Power Backoff.” *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, June 2002, pp. 1075-84.

N.P. Sands and K.S. Jacobsen. “Pilotless Timing Recovery for Baseband Multi-carrier Modulation.” In *IEEE Journal on Selected Areas in Communications*, vol. 20, no 5., June 2002, pp. 1047-54.

K.S. Jacobsen. “Methods of Upstream Power Backoff on Very High-Speed Digital Subscriber Lines.” In *IEEE Communications Magazine*, vol. 39, no. 3, March 2001, pp. 210-16.

J.M. Cioffi *et al.* “Very-high-speed Digital Subscriber Lines.” In *IEEE Communications Magazine*, vol. 37, no. 4, April 1999, pp. 72-79.

TECHNICAL CONFERENCE PAPERS

K.S. Jacobsen. “Design and Performance of Synchronized DMT (SDMT) Modems for VDSL.” In *ICCE Conference Record*, Los Angeles, CA, June 1999.

K.S. Jacobsen. “Synchronized DMT (SDMT) for Very high-speed Digital Subscriber Line (VDSL) Transmission.” In *Globecom '98 Conference Record*, Sydney, Australia, November 1998.

K.S. Jacobsen. “Discrete Multi-Tone Modulation for High-Speed Upstream Communications on HFC Networks.” In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 1997.

K.S. Jacobsen. “Synchronized Discrete Multi-Tone (SDMT) Modulation for Cable Modems: Making the Most of the Scarce Reverse Channel Bandwidth.” In *Wescon '97 Conference Proceedings*, Santa Clara, CA, November 1997.

K.S. Jacobsen and J.M. Cioffi. “Achievable Throughput of Multicarrier-based Multipoint-to-point Networks Using a Reservation-based Channel Access Protocol.” In *Globecom '96 Conference Record*, London, U.K., November 1996.

K.S. Jacobsen, J.A.C. Bingham and J.M.Cioffi. “Synchronized DMT for Multipoint-to-point Communications on HFC Networks.” In *Globecom '95 Conference Record*, Singapore, November 1995.

J.A.C. Bingham and K.S. Jacobsen. “Upstream Transmission in an HFC System using SDMT: the Network, Data Rates, and a MAC Protocol.” Presented at Globecom '95, Singapore, November 1995.

Krista S. Jacobsen

K.S. Jacobsen *et al.* “Very High Bit Rate Digital Subscriber Lines (VDSL): An Effective Deployment Strategy for FTTC Utilizing the Existing Copper Network.” In *Proceedings of VII International Workshop on Optical Access Networks*, Nuremberg, Germany, September 1995.

K.S. Jacobsen, J.A.C. Bingham and J.M. Cioffi. “A Discrete Multitone-based Network Protocol for Multipoint-to-point Digital Communications in the CATV Reverse Channel.” In *Canadian Cable Television Association (CCTA) Cablexpo Technical Papers*, Halifax, Nova Scotia, May 1995.

K.S. Jacobsen and J.M. Cioffi. “An Efficient Digital Modulation Scheme for Multimedia Transmission on the Cable Television Network.” In *Technical Papers, 43rd Annual National Cable Television Association (NCTA) Convention and Exposition*, New Orleans, LA, May 1994.

K.S. Jacobsen and J.M. Cioffi. “High-performance Multimedia Transmission on the Cable Television Network.” In *Proceedings 1994 International Conference on Communications*, New Orleans, LA, May 1994.

Patents

A.J. Redfern, G. Ginis, F.A. Mujica, and K.S. Jacobsen. “Spectrally flexible band plans with reduced filtering requirements.” U.S. Patent Number 7,342,937. March 2008.

K.S. Jacobsen, M.D. Agah, and B.R. Wiese. “Method to mitigate effects of ISDN off/on transitions in ADSL.” U.S. patent number 7,184,467. February 2007.

J.M. Cioffi, J.A.C. Bingham, and K.S. Jacobsen. “Method and apparatus for coordinating multi-point to-point communications in a multi-tone data transmission system.” U.S. patent number 7,110,370. September 2006.

J.M. Cioffi, J.A.C. Bingham, and K.S. Jacobsen. “Method and apparatus for coordinating multi-point to-point communications in a multi-tone data transmission system.” U.S. patent number 7,079,549. July 2006.

J.M. Cioffi, J.A.C. Bingham, and K.S. Jacobsen. “Method and apparatus for coordinating multi-point to-point communications in a multi-tone data transmission system.” U.S. patent number 7,068,678. June 2006.

J.M. Cioffi, J.A.C. Bingham, and K.S. Jacobsen. “Method and apparatus for coordinating multi-point to-point communications in a multi-tone data transmission system.” U.S. patent number 6,937,623. August 2005.

K.S. Jacobsen, B. Wiese, and C. Milbrandt. “Upstream power back-off.” U.S. patent number 6,922,448. July 2005.

J.M. Cioffi, J.A.C. Bingham, and K.S. Jacobsen. “Method and apparatus for coordinating multi-point to-point communications in a multi-tone data transmission system.” U.S. patent number 6,473,438. October 2002.

B. Wiese, K.S. Jacobsen, N.P. Sands, and J. Chow. “Initializing communications in systems using multi-carrier modulation.” U.S. patent number 6,434,119. August 2002.

K.S. Jacobsen and B. Wiese. “A method to mitigate the near-far FEXT problem.” U.S. patent number 6,205,220. March 2001.

J.A.C. Bingham and K.S. Jacobsen. “Methods for coordinating upstream discrete multi-tone data transmissions.” U.S. patent number 5,644,573. July 1997.

Prior Testimony

***IN RE MARRIAGE OF PRINCE* (SUP. CT. CAL., COUNTY OF SANTA CLARA, CASE NO.: 2012-1-FL-163041)**

Provided deposition and trial testimony on behalf of Petitioner, Jeffrey Prince.

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

TQ DELTA, LLC, v. 2WIRE, INC. Plaintiff, Defendant.	Civil Action No. 13-cv-1835-RGA
TQ DELTA, LLC, v. ZHONE TECHNOLOGIES, INC. Plaintiff, Defendant.	Civil Action No. 13-cv-1836-RGA
TQ DELTA, LLC, v. ZYXEL COMMUNICATIONS, INC. and ZYXEL COMMUNICATIONS CORPORATION, Plaintiff, Defendants.	Civil Action No. 13-cv-2013-RGA
TQ DELTA, LLC, v. ADTRAN, INC. Plaintiff, Defendant.	Civil Action No. 14-cv-954-RGA
ADTRAN, INC, v. TQ DELTA, LLC. Plaintiff, Defendant.	Civil Action No. 15-cv-121-RGA

**DECLARATION OF DR. CHRISSAN IN SUPPORT OF PLAINTIFF'S REPLY CLAIM
CONSTRUCTION BRIEF FOR FAMILY 6 PATENTS**

I, Douglas A. Chrissan, hereby declare under penalty of perjury:

1. I incorporate my declaration in support of Plaintiff's opening claim construction brief, served on September 20, 2017 ("Chrissan Opening Declaration" or "my Opening Declaration"), into this declaration.

2. In addition to the information disclosed in my Opening Declaration, I have also considered Defendants' answering claim construction brief ("Defendants' Answering Brief") and Dr. Krista Jacobsen's declaration in support of Defendants' answering claim construction brief ("Dr. Jacobsen's Answer Declaration"), served on October 18, 2017.

3. My declaration does not address Dr. Jacobsen's statements related to the issues of publication dates and legally proper incorporation by reference because these issues are not relevant to my opinions.

4. Defendants and Defendants' expert Dr. Jacobsen state that I rely on the disclosures of Digital Subscriber Line ("DSL") standards documents, specifically ITU-T Recommendations G.992.3 and G.993.1, in formulating my opinions of how a person of skill in the art would interpret the disputed claim terms. My opinions are not exclusively based on DSL technology or DSL standards. However, the inventors of the Family 6 patents described the embodiments of the inventions in terms of DSL. Therefore, it would be proper to look at the DSL technology and DSL standards when interpreting the inventions of the Family 6 patents. Furthermore, the perspective in which the claims should be interpreted by of a person of ordinary skill the art includes the perspective of a person of ordinary skill in DSL technology.

I. DISPUTED CLAIM TERMS

A. “transceiver”

5. Defendants reference the statement from the specification that “the components of the system can be located at any location within a distributed network without effecting the operation of the system” to justify an assertion that a transmitter and receiver need not be collocated. Defendants’ Answering Brief at p. 4 (citing A13 (‘835 patent) at 7:50–52, and Dr. Jacobsen’s Answer Declaration at ¶ 42). I disagree with this assertion for three reasons described below:

6. First, if the transmitter and receiver are not collocated, they are not in the same device; therefore, neither the device the transmitter is in nor the device the receiver is in can be considered a transceiver even under Defendants’ proposed construction.

7. Second, a person of skill would recognize that for a wired DSL transceiver, which uses a single subscriber line for transmission in both directions, the transceiver must *at least* include *some* analog front-end circuitry shared between its transmitter and receiver.

8. Third, a person of skill in the art would recognize that the statement “the components of the system can be located at any location within a distributed network without effecting the operation of the system” refers to a division of components that could, in fact, be distributed across a network without affecting the operation of the system. For example, certain messaging and management modules shown in Figure 1 of the Family 6 patents could be placed in different locations.

9. I note that Defendants omitted the sentence immediately following the cited statement above, which provides additional context:

As will be appreciated from the following description, and for reasons of computational efficiency, the components of the system can be arranged at any location within a distributed network without effecting the operation of the system. For example, the

various components can be located in a central office (CO or ATU-C) modem, a customer premises modem (CPE or ATU-R), or some combination thereof.

A13 ('835 patent) at 7:50–57 (portion omitted by the Defendants has been underlined). When read in context, the statement cited by the Defendants does not suggest that individual components of a transceiver can be broken up and arranged at any location. A13 ('835 patent) at 7:50–57. A transceiver is a well-known structure that the Family 6 patents do not attempt to redefine. Instead, the Family 6 specification refers to components of the “system” that can be arranged “at any location.” A13 ('835 patent) at 7:50–57. The “various components” that “can be located in a central office ... modem, or a customer premises ... modem ...” are for example, described by the Family 6 patents:

“a decoder/deinterleaver 210, a bit error rate (BER) detection module 220, a forward error correction and interleaving parameter (FIP) module 230, an impulse noise length determination module 240, an impulse noise period determination module 250, an impulse noise parameter management module 260, an impulse noise protection adaptation module 270, a synchronization module 280, a message module 290 and a controller and memory” in one modem or “a management module 310, a synchronization module 320, a message module 330, and may optionally include an impulse noise parameter management module 340” in another modem.

A13 ('835 patent) at 7:50–57, 8:4–20.

10. The dictionary definitions cited by Defendants do not undermine Plaintiff’s proposed construction. For example, the first dictionary cited by the Defendants provides these definitions:

Transceiver 1. Any device that transmits and receives. In sending and receiving information, it often provides data packet collision detection as well.

2. In IEEE 802.3 networks, the attachment hardware connecting the controller interface to the transmission cable. The transceiver contains the carrier-sense logic, the transmit/receive logic, and the collision-detect logic.

A160 (Newton’s Telecom Dictionary, 2004) at 846 (underline emphasis added). The first of these definitions places the transmitting and the receiving function in a single device and states

“[i]n sending and receiving information, it often provides data packet collision detection as well.” *Id.* A person of ordinary skill in the art would understand that data packet collision detection involves a shared communication medium (e.g., a wire or cable used for both transmit and receive) and involves coordination between a transmitter and receiver. Both of these features require at least some shared circuitry between the transmitting and receiving functions.

11. Another dictionary cited by the Defendants also supports Plaintiffs’ construction:

Transceiver A device that can both transmit and receive signals on a communication medium. Many communication devices, including *modems, *codecs, and *terminals, are transceivers.

A163 (Oxford Dictionary of Computing, 2004) at 542. Again, the definition references a single device. It also states that transmission and reception occur over a single, i.e., shared, communication medium. Furthermore, one of skill in the art understands that the modem and codec mentioned as examples mean modulator/demodulator and coder/decoder, which indicate shared circuitry (e.g., clock circuitry or signal processing circuitry) between the modulator and demodulator, and coder and decoder, respectively.

12. My interpretation of a “transceiver,” in the context of the Family 6 patents, is the same independent of the communication channel, or whether the transceiver is for a wired network or a wireless network. Furthermore, a person of skill in the art would recognize that the dictionary definitions cited in my Opening Declaration apply to wired as well as wireless transceivers. Chrissan Opening Declaration at ¶ 42. For example, the person of skill would recognize that for a wired DSL transceiver, which uses a single subscriber line for transmission in both directions, the transceiver must *at least* include *some* analog front-end circuitry shared between its transmitter and receiver.

B. “flag signal” for ’835 patent, claims 8 and 10

13. As described in Paragraph 46 of my Opening Declaration, the Family 6 patent specification describes two primary embodiments for synchronizing the switch in FIP or interleaver parameter settings: synchronization using Forward Error Correction (“FEC”) codeword counters, or using a flag signal. A15 (’835 Patent) at 11:4–9 (“the receiver and transmitter can synchronize the modification of the FEC and interleaving parameters such that the both the transmitter and receiver start using the parameters at the same instant in time. This synchronization can be based on, for example, a synchronization using FEC codeword counters or a flag signal.”). In the first embodiment for synchronization the switch occurs on a particular FEC codeword count value contained in a message. A15 (’835 Patent) at 11:16–20 (“a message indicating the FEC codeword count value on which the FIP parameters will be updated.”); *id.* at 11:55–57 (“[f]or example, the message can indicate that when the codeword counter equals 501, the new FIP setting will be used for transmission and reception.”). The second embodiment for synchronization is based on a flag signal. A15 (’835 Patent) at 12:8–11 (“For synchronization using a flag signal, the receiver and transmitter would start using updated FEC and interleaving parameters on a pre-defined FEC codeword boundary following the sync flag.”). Defining a “flag signal” as an “inverted sync symbol or sync flag used to synchronize the switch to using an updated FIP setting” properly excludes a message-based embodiment while properly encompassing the flag signal embodiment.

14. Defendants take the phrase “Flag Signal indicating when updated FIP Parameters are to be used” from the Family 6 patents at S630 of Fig. 6 and conclude that this means a “flag signal” is a “signal indicating when updated FIP settings are to be used.” A8 (’835 patent) at S630 of Fig. 6. This definition removes the concept of a “flag” being a simple marker that is either present or not present, as the inventors describe by referring a person of ordinary skill in

the art to the G.992.3 ADSL2 standard for an example of a flag signal. A “flag signal” is a certain type of signal: one that either is present or not present, but otherwise does not carry additional information. Defendants’ proposed additional phrase “indicating when updated FIP parameters are to be used” is a purpose of the flag signal, but even if a construction were synthesized by adding that phrase to “flag signal” (*i.e.*, if “flag signal” were to be construed as “flag signal indicating when updated FIP settings are to be used”), that construction would not sufficiently clarify the meaning of “flag signal.” Also, Fig. 6 of the ‘835 patent and its related description describe at a high level a “method of synchronization using a flag signal,” rather than what a flag signal is. *Id.*

15. Contrary to Defendants’ assertion, an “inverted sync symbol” and “sync flag” are terms one of skill in the art would understand at least because the Family 6 patents point to the specific portion of the G.992.3 standard that explains what is meant by these terms. *See* A15 (‘835 Patent) at 12:29–31 (“For example, the flag signal could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol.”). The G.992.3 (07/2002) recommendation explains that the “transmit function inserts a synchronization symbol every 68 data symbols” and that “[e]very time the transmit PMD function receives a PMD.Synchflag.request primitive (related to on-line reconfiguration during the L0 state) from the transmit PMS-TC layer, the phase of the first next inserted synchronization symbol shall be inverted.” A120 (G.992.3 (07/2002)), § 8.7.3. Furthermore, it also describes “[a]t the start of Showtime, the first synchronization symbol transmitted shall be an SS-REVERB symbol,” and that synchronization symbols are distinct from data-carrying symbols. *Id.* at §§ 8.7.2 & 8.7.3.

16. The G.992.3 (07/2002) document also describes that the SS-REVERB signal is generated by modulating a Pseudo Random Binary Sequence (“PRBS”), *i.e.* a known bit pattern,

onto the subcarriers of the transmitted signal to generate a known DMT signal. A119 (G.992.3 (07/2002)), § 8.7.1. Furthermore, an inverted sync symbol (that is inverted relative to an SS-REVERB signal) is an “SS-SEGUE” symbol, defined as “a subcarrier-by-subcarrier 180 degrees phase reversal of an SS-REVERB symbol (i.e., an SS-SEGUE symbol modulates the bitwise inverted REVERB PRBS data pattern).” *Id.* at § 8.7.

17. Therefore, a person of skill in the art, directed to the G.992.3 (07/2002) standards document by the above quoted statement in the Family 6 patents, would not “have had to guess the meaning of an ‘inverted sync symbol’ or ‘sync flag,’” but would instead look to the G.992.3 standard for a definition of these terms. Consequently, a person of skill in the art would readily understand that “the phase of the first next inserted synchronization symbol shall be inverted” in G.992.3 describes an inverted sync symbol and that a “PMD.Synchflag” primitive and/or the “line signal corresponding to a PMD.Synchflag primitive” in the G.992.3 standard describes a “sync flag.” *See* A15 (’835 Patent) at 12:29–31, A231–36 (G.992.3 (07/2002)) at § 9.4.1 and Table 9-2.

18. The term “flag signal” is used in the specification to refer to the “inverted sync symbol” and “sync flag” terms. *See, e.g.*, A15 (’835 Patent) at 12:8–11 (“For synchronization using a **flag signal**, the receiver and transmitter would start using updated FEC and interleaving parameters on a pre-defined FEC codeword boundary following the **sync flag**.”) and 12:29–30 (“For example, the flag signal could be an inverted sync symbol, or sync FLAG . . .”).

19. The term “sync flag” in the Family 6 patents, and the PMD.Synchflag primitive of G.992.3, correspond to an event or marker that either happens or does not happen with respect to a given modem symbol. As described by G.992.3, a PMD.Synchflag primitive is associated with a line signal, which in G.992.3 is an inverted sync symbol. However, one of skill the art

understands that—outside of G.992.3 but still within the scope of the Family 6 patents—other line signals may be used to indicate a sync flag so long as the utilized line signal is a marker or flag that either occurs or does not occur, but when it occurs it does not carry additional information within it. Plaintiff’s proposed construction of “flag signal” supports this correct interpretation of flag signal.

20. Therefore, the meaning of “flag signal” is readily understood by one of skill in the art both because of the examples given in the specification and because the term has a plain meaning that is consistent with these examples. The word “flag” as used in the term “flag signal” is a term of art that is used to refer to a recognizable signal or bit pattern that serves as a marker for synchronization. Technical dictionaries from the relevant timeframe confirm this:

- flag (1) A two-state indicator/variable such as yes/no, true/false, on/off, error/no-error, and ready/not-ready. It is a variable set at one point in a program for use later in the same or different programs. It informs these programs that a condition has been met or not met. (2) In synchronous transmission systems, the *flag* is a specific bit pattern that marks the beginning and end of a packet (frame). See also *flag character*...

A251–54 (Hargrave’s Communications Dictionary (2001)) at 214.

- Flag... 2. In synchronous transmission, a flag is a specific bit pattern (usually 01111110) used to mark the beginning and end of a “frame” of data....

A255–57 (Newton’s Telecom Dictionary, 19th ed., March 2003) at 328. These definitions of flag signal are consistent with the inverted sync symbol and sync flag referenced in the Family 6 patents.

21. Defendants also argue that “Plaintiff attempts to import into the claims transmitter and receiver synchronization that is simply not required... The asserted claims are directed only to actions of the transmitter portion of the transceiver that transmits the flag signal to the receiver portion of an (unclaimed) transceiver... Stated another way, the asserted claims include only

limitations governing the transmitting portion of one transceiver.” Defendants’ Answering Brief § 5.B.4, pp. 10–11. I agree that “the asserted claims include only limitations governing the transmitting portion of one transceiver” and that the asserted claims are directed to a transmitting transceiver and its transmission of the flag signal. Plaintiff’s proposed construction of flag signal characterizes what a “flag signal” is partially in terms of how it is used, but it does not recite elements or actions of the receiver.

22. The specification describes the transmitting transceiver transmitting a flag signal that is used to synchronize a change and a receiving transceiver using the flag signal to synchronize a change:

The transmitting modem then sends a flag or marker signal to the receiving modem 200 indicating that the new FIP settings are to be used on a predetermined number of DMT symbols following the transmission of the flag or marker signal. . . . The transmitting modem 300 then starts using the new FIP settings for transmission on the predetermined number of DMT symbols following the transmission of the flag or marker signal. Similarly, the receiving modem starts using the new FIP settings for reception once the predetermined number of DMT symbols following the receipt of the flag or marker signal have been received.

A15 (’835 Patent) at 12:25–37. Plaintiff’s proposed construction correctly defines the flag signal and does not recite elements or actions of the receiving transceiver.

C. “flag signal” for ’162 patent, claims 8 and 9

23. The difference between the proposed construction for the term “flag signal” in the context of the ’835 patent and the ’162 patents is that the asserted claims of the ’835 patent refer to an FIP setting, and the asserted claims of the ’162 patent refer to an interleaver parameter.

24. For the same reasons discussed in Section I.B above, a “flag signal” in the context of the ’162 patent means an “inverted sync symbol or sync flag used to synchronize the change to using an updated interleaver parameter value.”

D. “interleaver parameter value”

25. Defendants state that the parties disagree on “whether the interleaver parameter value refers to an interleaver depth.” Defendants’ Answering Brief at § V.D, p. 13. In my opinion both Plaintiff’s proposed construction (“numerical value of an interleaver depth parameter”) and Defendant’s proposed construction (“the numerical value of the interleaver depth in number of codewords”) *refer* to an interleaver depth. However, Defendants’ dropping of the word “parameter” from their construction blurs the meaning of the term. Furthermore, as explained in my Opening Declaration and below, the addition of “in number of codewords” in their proposed construction is incorrect.

26. The parties disagree on whether the additional phrase “in number of codewords” is necessary or accurate. It is not necessary, because a person of skill in the art would understand “interleaver parameter value” to be the numerical value of an interleaver depth parameter independent of any further clarification. It is not accurate because the interleaver parameter value is simply a number that defines how the byte position of bytes of the input blocks of an interleaver will be delayed (and therefore, spaced apart) in the interleaved output. While Defendants’ expert Dr. Jacobsen initially states in paragraph 66 of her declaration that she disagrees with this point, she immediately makes a contrary statement in paragraph 67 that is in substantive agreement with me on this point. *Compare* Dr. Jacobsen’s Answer Declaration at ¶ 66 (“a person having ordinary skill in the art at the time of the alleged invention would not have understood the term ‘interleaver depth’ to be ‘a number that defines, per the DSL standards, how the byte position of bytes of the input blocks will be delayed in the interleaver output.’”) with ¶ 67 (“[a] skilled artisan would have understood that the interleaver depth is one plus the minimum separation, in bytes, at the output of the interleaver between any two bytes that were adjacent at the input of the interleaver. *See, e.g.,* Cioffi Course Reader, Chapter 11 at 258

(available at <https://web.stanford.edu/group/cioffi/ee379b/reader.html>).”) Notably, Dr. Jacobsen’s definition of interleaver depth cited above does not require or even mention that the interleaver depth be expressed in a number of codewords. Rather, it would be more appropriate to consider interleaver depth as being expressed in a number of bytes.

27. An interleaver depth parameter value, whether expressed in codewords (as Defendants construction requires), fractions of codewords, bytes, symbols, or another unit, is still an interleaver depth parameter value as understood by one of skill in the art. Even Defendants’ own expert, Dr. Jacobsen, when providing a definition for an interleaver depth expressed in a number of “bytes,” cites to a reference that provides a definition for interleaver depth expressed in symbol periods. *See* Dr. Jacobsen’s Answer Declaration at ¶ 67 (citing Cioffi Course Reader, Chapter 11); *see also* A276 (Cioffi Course Reader, Chapter 11) at 263 (“Definition 11.2.1 (Depth of an Interleaver) The depth J of an interleaver is defined as the minimum separation in **symbol periods** at the output of the interleaver between any two symbols that were adjacent at the input of the interleaver.”) (emphasis added).

28. I also note that Defendants cite to the background section of the Family 6 patents to support their argument that interleaver depth must be defined in codewords. The cited background section states “ D is the interleaver depth in codewords.” Defendants’ Answering Brief at p. 14 (citing A10 (‘835 patent) at 2:13–14). Yet, when this statement is considered in context, it becomes clear that the specification is describing a characteristic of “ D ” from one particular DSL standard—G.992.3—as opposed to providing a definition of an interleaver depth parameter:

Impulse noise protection is defined in the ADSL2 Standard G.992.3, which is incorporated herein by reference in its entirety, as the number of impulse noise corrupted DMT symbols that can be corrected by the FEC and interleaving configuration. Specifically, G.992.3 defines the following variables:

$$\begin{aligned} \text{INP} &= 1/2 * (S * D) * R / N \\ S &= 8 * N / L \\ \text{Latency (or delay)} &= S * D / 4 \\ \text{Line Rate (in kbps)} &= L * 4 \end{aligned}$$

where N is the codeword size in bytes, R is the number of parity (or redundancy) bytes in a codeword, D is the interleaver depth in number of codewords, and L is the number of bits in a DMT symbol.

A10 ('835 patent) at 1:65 – 2:15 (emphasis added). A person of ordinary skill in the art would understand that the import of this example provided in the background section of the patent is that the ADSL2 standard uses a particular interleaver depth parameter, “D,” that can be expressed in a number of codewords. The reason for this is that the ADSL2 standard uses an interleaver input block length that is equal to the FEC codeword length.

29. However, as explained throughout this section, and in my Opening Declaration, the interleaver input block length is not required to be equal to an FEC codeword in length. It was recognized by those of skill in the art prior to the filing of the Family 6 patents that the interleaver input block size could be an integer sub-multiple of a codeword length in bytes. If the interleaver input block is an integer sub-multiple of a codeword, this means that the interleaver block length is 1/N times the codeword length, where N is an integer and the interleaver block length is still an integer too. For example, if the codeword length is 128 bytes, the interleaver input block length could be 64 bytes or 32 bytes, which is one-half of the codeword length or one-fourth of the codeword length, respectively. Also, the book DSL Advances (2003) by Thomas Starr, Massimo Sorbara and John M. Cioffi explains that for VDSL, “[t]riangular interleaving [i.e., convolutional interleaving] that allows interleaving at a block length that is any integer sub-multiple of a code word length (in bytes) is allowed (ADSL forced the block size of the interleaver to be equal to the code word length).” A200 (Thomas

Starr et al., *DSL Advances* (2003)) at 226. In this case, the interleaver block length is the codeword length divided evenly by an integer N , such that the interleaver block length is still an integer.

30. I also note that nothing in the Family 6 patents excludes using an interleaver that uses an input block size that is an integer sub-multiple of a codeword. On the contrary, this is exactly the scheme employed by the ITU-T VDSL G.993.1 standard referenced in the Family 6 patents. See A129 (ITU-T G.991.3 (06/2004)) at § 8.3 *Forward error correction* (“A Reed-Solomon code word contains $N = K + R$ bytes ...”) and *Table 8-1/G.993.1 – Characteristics of convolutional interleaving* (identifying the parameter “Interleaver block length (I)” as having a value of “ I bytes (equal to or divisor of N).”). It is also the scheme used in the ETSI TS 101 270-2 V1.2.1 standard. A204-205 (ETSI TS 101 270-2 V1.2.1 (2003)) at § 6.4.3 (“Interleaving”) (“[t]he convolutional interleaver is defined by two parameters: the interleaver block length, I , and the interleaving depth, D . The block length I divides the RS codeword length N .”), and Table 19 (describing “Interleaver block length” parameter as having the value of “ I octets [i.e., bytes] (I must divide N)”). Because in these examples the interleaver block size is not a whole codeword, it would be more appropriate to think of interleaver depth as being expressed in fractions of codewords or in a number of bytes.

31. Defendants state that “[a] person of ordinary skill in the art would not have necessarily thought of convolutional interleaving with respect to the Family 6 patents as it is not mentioned in the specification.” Defendants’ Answering Brief at p. 2, citing Dr. Jacobsen’s Answer Declaration at ¶ 31. I disagree. A person of ordinary skill in the art would have necessarily thought of convolutional interleaving with respect to the Family 6 patents for two reasons: 1) there were only two main kinds of interleaving in use at the time, “block” and

“convolutional,” and 2) DSL standards that used an interleaver up to the time of the Family 6 patents (e.g., ITU-T G.992.1 (06/99), ITU-T G.992.3 (07/2002), G.993.1 (06/2004), and ETSI TS 101 270-2 (2003)) used convolutional interleaving¹. See A220–21 (ITU-T G.992.1 (06/99)) at § 7.6.3, A118 (ITU-T G.992.3 (07/2002)) at § 7.7.1.5, A130–32 (G.993.1 (06/2004)) at § 8.4; A204–205 (ETSI TS 101 270-2 (2003)) at § 6.4.3. See also A271 (Cioffi Course Reader, Chapter 11) at 258 (describing “three popular classes of interleaving methods: block, convolutional, and random,” wherein “random” is described in Section 11.2.4 as a variation of a block interleaver). In addition to the fact that convolutional interleavers were consistently used in the DSL standards, Defendants’ own expert Dr. Jacobsen has previously explained that convolutional interleavers were preferred over other types for DSL applications. See A258–69 (Philip Golden, Herve Dedieu and Krista S. Jacobsen, *Fundamentals of DSL Technology*) at § 9.4 (where on p. 259 (A262) Dr. Jacobsen’s book describes block and convolutional interleavers, and states “[b]ecause block interleavers do not lend themselves to low-memory, efficient implementations, many DSL systems use convolutional interleaving...”). Therefore, a person of ordinary skill in the art would understand the disclosure of interleaving in the Family 6 patents in the context of convolutional interleavers.

32. In summary, based on Dr. Jacobsen’s, Dr. Cioffi’s and my explanations above of interleaver depth, and other known examples of DSL interleavers where it would be inappropriate to limit interleaver depth to being expressed in number of codewords, it is incorrect to categorically define an interleaver parameter value as an interleaver depth parameter value

¹ The optional “data interleaving” in G.991.2 (02/2001) (“SHDSL”) relates to allocating a single data stream to two separate twisted-pair connections, an operation unrelated to the Family 6 patents.

that is in a number of codewords. The correct definition is “numerical value of an interleaver depth parameter.”

E. “FIP setting”

33. As stated in Paragraph 54 of my Opening Declaration, an “FIP setting” (which both Defendants and their expert Dr. Jacobsen agree stands for “FEC and Interleaving Parameters setting,” *see* Dr. Jacobsen’s Answer Declaration at ¶ 72) is a set including at least one forward error correction parameter value and at least one interleaver parameter value.

34. It is correct to include “parameter value” in the construction because the claims recite switching from a “first FIP setting” to a “second FIP setting.” *See, e.g.*, A20 (’835 patent) at Claim 8. Without the word “value” in the construction, the change from a first FIP setting to a second FIP setting could be confused as requiring a switch from a first set of parameters, e.g., parameters A and B, to a different set of parameters, e.g., parameters B and C. Including the word “value” in the construction clarifies that it is the change in value of the parameters that distinguishes the first FIP setting from the second FIP setting. For example, the claim would cover a first FIP setting where the value of parameter A = 5 and parameter B = 20 and a second FIP setting where the value of parameter A = 5 and parameter B = 35.

35. Using a television analogy, picture quality parameter settings might be expressed as follows:

Setting 1

Parameter	Parameter Value
Brightness	+15
Contrast	-3

Setting 2

Parameter	Parameter Value
-----------	-----------------

Brightness	+15
Contrast	+5

36. In the context of the Family 6 patents, example FIP settings may be illustrated as follows:

First FIP Setting

Parameter	Parameter Value
codeword size (example FEC parameter)	192
interleaver depth (example interleaver parameter)	32

Second FIP Setting

Parameter	Parameter Value
codeword size (example FEC parameter)	168
interleaver depth (example interleaver parameter)	16

37. It can be easily understood from the above examples why omitting the word “value” from the construction incorrectly drops the concept of a “setting.” Indeed, the claims themselves confirm this. For example, claim 8 of the ’835 patent, reads “first FIP setting comprises at least one first FIP value,” and “second FIP setting comprises at least one second FIP value.” A20 (’835 patent) at Claim 8. Thus, an “FIP setting” as used in the claims requires that it be expressed in terms of parameter values and not just parameters.

F. “FIP value”

38. As stated in Paragraph 56 of my Opening Declaration, an “FIP value,” or “FEC and Interleaving Parameters value,” is a numerical value of a forward error correction parameter or numerical value of an interleaver parameter. I believe the parties’ dispute over “FIP value” is essentially rooted in their dispute over “FIP setting” above; therefore, my discussion for “FIP setting” also applies to this term “FIP value.”

G. “the switching occurs on a pre-defined forward error correction codeword boundary”

39. As stated in my Opening Declaration, the term “pre-defined forward error correction [FEC] codeword boundary” conveys sufficient meaning to a person of ordinary skill art at the time of the invention of the Family 6 patents. Furthermore, the term “the switching occurs on a pre-defined forward error correction codeword boundary” would be understood by a person of ordinary skill in the art as “the switching to an updated FIP setting is effective on the boundary of a forward error correction codeword where the position of the boundary of each codeword is known prior to the switching.”

40. The full claim element in which the disputed term appears reads: “the switching occurs on a pre-defined [FEC] codeword boundary following the flag signal.” A person of ordinary skill in the art would understand that this limitation is met when the switching occurs on any one of the pre-defined FEC codeword boundaries that follow the flag signal. The FEC codeword boundary upon which the switch occurs is thus limited to being a codeword boundary that is known in advance of the switch and one that follows a flag signal. There is nothing in that term that requires the switch to occur on any one particular pre-defined FEC codeword boundary. Further, a person of ordinary skill in the art would understand that if the switching occurs on a pre-defined FEC codeword boundary preceding the flag signal, the claim limitation would not be met. Similarly, it would be understood that the claim is not met if the switching did not occur on a codeword boundary that was not defined prior to the switch.

H. “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary”

41. As described in Paragraph 63 of my Opening Declaration, and similar to my discussion in Section I.G above, a person of skill in the art would understand “the second

interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary” to mean “the second interleaver parameter value is used for transmission starting on a boundary of a forward error correction codeword where the position of the boundary of each codeword is known prior to use of the second interleaver parameter value.”

42. I declare under penalty of perjury of the law of the United States that the foregoing is true and correct.

Date: November 09, 2017


Dr. Douglas A. Chrissan

**UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

<div>TQ DELTA, LLC,</div> <div>Plaintiff,</div> <div>v.</div> <div>2WIRE, INC.,</div> <div>Defendant.</div>	Civil Action No. 13-cv-1835-RGA
<div>TQ DELTA, LLC,</div> <div>Plaintiff,</div> <div>v.</div> <div>ZYXEL COMMUNICATIONS, INC.,</div> <div>and</div> <div>ZYXEL COMMUNICATIONS CORPORATION,</div> <div>Defendants.</div>	Civil Action No. 13-cv-2013-RGA
<div>TQ DELTA, LLC,</div> <div>Plaintiff,</div> <div>v.</div> <div>ADTRAN, INC.,</div> <div>Defendant.</div>	Civil Action No. 14-cv-954-RGA
<div>ADTRAN, INC.,</div> <div>Plaintiff,</div> <div>v.</div> <div>TQ DELTA, LLC,</div> <div>Defendant.</div>	Civil Action No. 15-cv-121-RGA

**DECLARATION OF DR. KRISTA S. JACOBSEN IN SUPPORT OF
DEFENDANTS' FAMILY 6 SUR-REPLY CLAIM
CONSTRUCTION BRIEF**

I. INTRODUCTION

1. I am providing this declaration in support of Defendants' Sur-Reply Claim Construction Brief for the Family 6 Patents.

2. In addition to the materials I reviewed to prepare my declaration in support of Defendants' claim construction brief ("my previous declaration" or "Jacobsen Response Dec."), which included "Plaintiff's Opening Claim Construction Brief For Family 6" ("TQ Delta Opening Brief") and the "Declaration of Dr. Douglas Chrissan In Support of TQ Delta's Family 6 Claim Construction Brief" ("Chrissan First Dec."), in preparing this declaration I have also reviewed "Plaintiff's Reply Claim Construction Brief For the Family 6 Patents" ("TQ Delta Reply Brief") and the "Declaration of Dr. Chrissan In Support of Plaintiff's Reply Claim Construction Brief For Family 6 Patents" ("Chrissan Reply Dec.").

3. Dr. Chrissan downplays his heavy reliance on various DSL standards in his previous declaration, now stating the "the perspective in which the claims should be interpreted by a person of ordinary skill in the art includes the perspective of a person of ordinary skill in DSL technology." Chrissan Reply Dec. at ¶ 4. Yet in his previous declaration, Dr. Chrissan did not opine that a person having ordinary skill in the art worked in or had experience in DSL. Chrissan First Dec. at ¶ 17. Instead, Dr. Chrissan opined that "a person of skill in the art would be a person with . . . at least 3 years of experience working with such multicarrier systems." Chrissan First Dec. at ¶ 17. At the time of the alleged invention, multicarrier modulation, including DMT and OFDM, was used in a variety of environments outside of DSL, including, for example, digital audio broadcasting. Therefore, in my opinion, it is improper for Dr. Chrissan to support his claim construction by making arguments that are not supported by his own definition of a person of ordinary skill in the art.

4. TQ Delta states that it relied, in its opening brief, on “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” only to provide “background information.” TQ Delta Reply Brief at 3. But TQ Delta relied heavily on the contents of G.992.3 to explain its own proposal for the term “flag signal,” including what it contends the terms “sync flag” and “inverted sync symbol” in its proposal mean.

5. Moreover, neither TQ Delta nor Dr. Chrissan responded to my observation that the descriptions “the ADSL series of ITU G.992.x standards and the VDSL series of ITU G.993.x standards” and “the ADSL2 Standard G.992.3” do not adequately identify the documents purportedly incorporated by reference because the sets of documents those descriptions identify change as time passes.

II. DISPUTED CLAIM TERMS

A. “transceiver”

6. It remains my opinion that a person having ordinary skill in the art at the time of the alleged invention would have understood “transceiver” to mean “communications device capable of transmitting and receiving data” as Defendants propose.

7. Dr. Chrissan and TQ Delta contend that a “person of skill would recognize that for a wired DSL transceiver, which uses a single subscriber line for transmission in both directions, the transceiver must *at least* include *some* analog front-end circuitry shared between its transmitter and receiver.” TQ Delta Reply Brief at 6; Chrissan Reply Dec. at ¶ 7 (emphasis in originals). I disagree. As a person having ordinary skill in the art at the time of the alleged invention would have understood, one could use a splitter upstream of a transceiver to provide separate physical paths for the transceiver’s transmitter portion and its receiver portion. There is no inherent requirement for any transceiver to include any circuitry shared between its

transmitter and receiver, much less for a DSL transceiver to include shared analog front-end circuitry.

8. Furthermore, TQ Delta suggests that Defendants’ proposed construction does not require the transmitter and receiver portions of a transceiver to reside in the same physical enclosure. TQ Delta Reply Brief at 7-8. As I explained previously, a skilled artisan would understand Defendants’ construction, which recites a single “communications device,” to include both the transmitter and receiver portions of the transceiver.

9. Dr. Chrissan and TQ Delta also argue that a dictionary definition stating that the transmitting and receiving functions are in a single device, and a dictionary definition stating that modems, codecs, and terminals are transceivers, lead to the conclusion that transceivers must include shared circuitry. TQ Delta Reply Brief at 7-8; Chrissan Reply Dec. at ¶¶ 10-11. I disagree. As a skilled artisan would have understood, and as Dr. Chrissan and TQ Delta point out, the terms “modem” and “codec” are simply shorthand for “modulator/demodulator” and “coder/decoder.” TQ Delta Reply Brief at 8; Chrissan Reply Dec. at ¶ 11. In my opinion, neither term would have led a skilled artisan to the conclusion that transceivers necessarily include shared circuitry.

B. “flag signal” (claims 8 and 10 of the ’835 patent)

10. TQ Delta and Dr. Chrissan agree with the Defendants and me that the Family 6 patents disclose two embodiments, and that both disclosed embodiments enable the transmitter in a first transceiver and the receiver in a second transceiver, in communication with the first transceiver over a communication channel, to switch from a first FIP setting to a second FIP setting at the same instant in time, *i.e.*, to synchronize their transitions to the second FIP setting. ’835 patent at col 11:4-9; TQ Delta Reply Brief at 9; Chrissan Reply Dec. at ¶ 13. A key difference between the two disclosed embodiments is the mechanism by which one of the

transceivers informs the other of the point in time at which it will begin using the second FIP setting: one embodiment uses an FEC codeword counter value, and the other uses a flag signal.

11. TQ Delta and Dr. Chrissan refer to the embodiment using an FEC codeword counter as the “message-based embodiment,” which is misleading because both disclosed embodiments use at least one message, namely, a message that one transceiver sends to the other transceiver to convey the second FIP setting. ’835 patent at col. 11:39-46; col. 12:17-24; FIG. 3 (box S370); FIG. 5 (step S530 of FEC codeword counter embodiment is “Receiver sends to Transmitter a message with the updated FIP Parameters or the transmitter sends to the Receiver a message with the updated FIP Parameters”); FIG. 6 (step S620 of flag signal embodiment is “Exchange Message for updated FIP Parameters”).

12. For clarity, I will refer to the two embodiments as the “FEC codeword counter embodiment” and the “flag signal embodiment.”

1. The FEC codeword counter embodiment

13. The FEC codeword counter embodiment is illustrated in FIG. 5 (copied below; *see* blocks S530 and S540 in particular) and described in the ’835 patent at column 11, lines 10-65 and at column 18, line 47 through column 19, line 14.

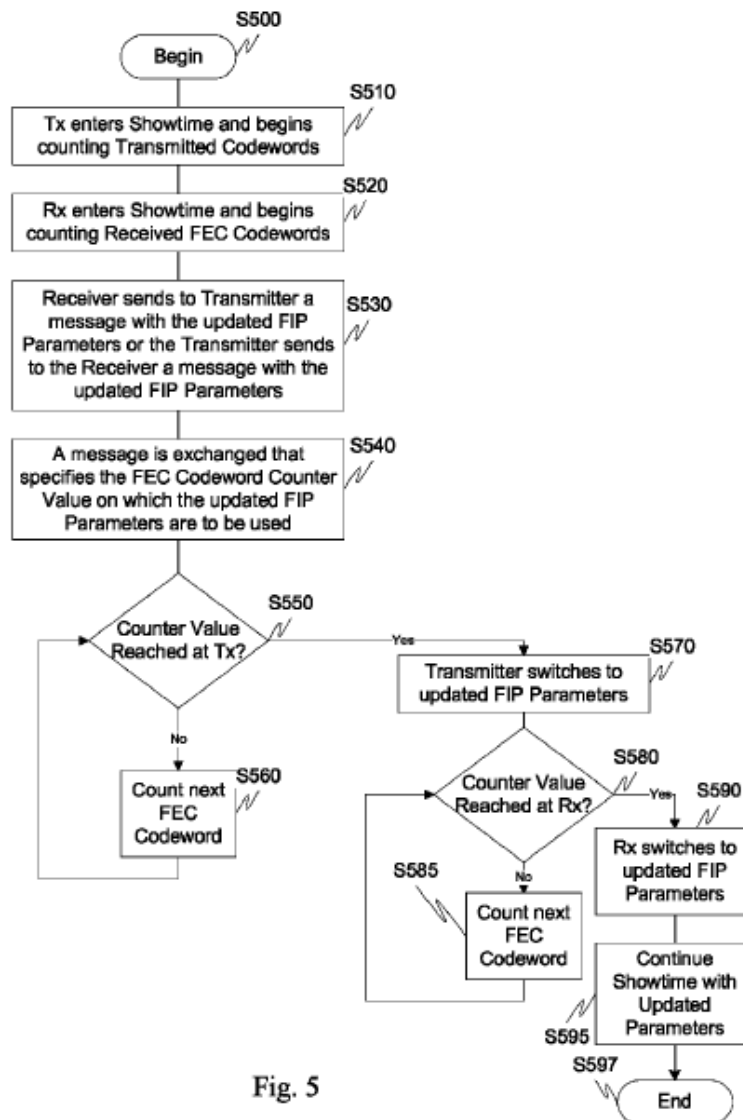


Fig. 5

14. In the FEC codeword counter embodiment, the transmitting transceiver and receiving transceiver both maintain FEC codeword counters, which are incremented by 1 for each transmitted (or received) FEC codeword. '835 patent at col. 11:10-16. The counters are synchronized with each other, meaning that, at the transmitter, the FEC codeword corresponding to the transmitter's FEC codeword counter value being the integer n is the same codeword as the FEC codeword at the receiver corresponding to the receiver's FEC codeword counter value being the integer n . *Id.* at col. 11:20-38. In other words, the transmitter's and receiver's FEC

codeword counters identify the same individual FEC codewords in the sequence of FEC codewords transmitted by the transmitter and received by the receiver.

15. To initiate a change from a current FIP setting to a new FIP setting, transceiver A sends a first message to transceiver B, where A can be the transmitting transceiver or the receiving transceiver, and B is whichever transceiver A is not. '835 patent at col. 11:10-65; FIG. 5 (step S530); col. 18:47-19:14. This first message tells transceiver B that a change to a new FIP setting is necessary and provides the new FIP setting. *Id.* at col. 11:39-46; col. 18:53-60.

16. A second, separate message, which could be sent by either transceiver A or transceiver B, tells the other transceiver the value of the FEC codeword counter at which both the transmitter and the receiver will switch from using the current FIP setting to using the new FIP setting. '835 patent at col. 11:16-20; col. 11:47-57; col. 18:61-63; FIG. 5 (step S540). Transceivers A and B then both switch to the new FIP setting when their respective FEC codeword counters reach the specified value. *Id.* at col. 11:58-65; col. 18:63-11; FIG. 5 (steps S570, S590).

17. Because the FEC codeword counters in the transmitter and in the receiver are synchronized, by conveying in the second message the value of the FEC codeword counter on which the switch will occur, the second message ensures that both transceivers have the necessary information to enable them to switch to the new FIP setting at the same location in the stream of FEC codewords. Thus, the switch occurs without the connection dropping and the transceivers having to reinitialize, something that the patents emphasize is desirable to avoid. *See, e.g.*, '835 patent at col. 2:64-67; col. 3:37-47; col. 9:19-29.

2. The flag signal embodiment

18. The flag signal embodiment is illustrated in FIG. 6 (copied below; *see* blocks S620 and S630 in particular) and described at column 11, line 66 through column 12, line 37 and at column 19, lines 15-30 of the '835 patent.

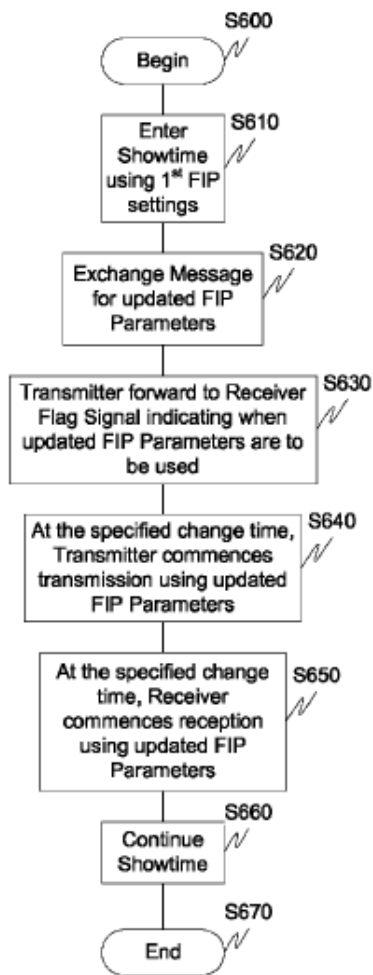


Fig. 6

19. The Family 6 patents present the flag signal embodiment as an alternative to the FEC codeword counter embodiment, and TQ Delta acknowledges that it is a distinct embodiment. '835 patent at col. 11:4-9; TQ Delta Reply Brief at 9. The Family 6 patents state

that the flag signal embodiment “may be more desirable than using an FEC codeword counter because, for example, it has greater impulse noise immunity,” (’835 patent at col. 12:5-7), which a skilled artisan would have understood to mean that the transceivers do not use FEC codeword counters in the flag signal embodiment.

20. In the flag signal embodiment, transceiver A sends only one message to transceiver B (where A can be the transmitting transceiver or the receiving transceiver, and B is whichever transceiver A is not). The message tells transceiver B that a change to a new FIP setting is necessary and provides the new FIP setting. ’835 patent at col. 12:17-24; col. 19:19-20; FIG. 6 (step S620).

21. Unlike in the FEC codeword counter embodiment, in the flag signal embodiment neither transceiver provides to the other transceiver an FEC codeword counter value to synchronize the transceivers’ switch to the new FIP setting. Instead, the transmitting transceiver sends a flag signal to synchronize the transceivers’ switch to the new FIP setting. ’835 patent at col. 12:25-29; col. 19:20-22; FIG. 6 (step S630). The flag signal “indicat[es] when the new FIP settings are to be used.” ’835 patent at col. 19:21-22. Defendants’ construction is consistent with such disclosure; and although it would have been clear to a person having ordinary skill that a flag signal encompasses an “inverted sync signal,” “flag,” and/or “marker signal,” it would not have been clear to a person having ordinary skill that the same can be said of an inverted sync symbol or sync flag, as TQ Delta proposes.

22. Thus, the flag signal embodiment differs from the FEC codeword counter embodiment in three ways. Specifically, in the flag signal embodiment: (1) neither transceiver provides an FEC codeword counter value to the other to synchronize the switch to the new FIP

setting, (2) the transceivers do not use FEC codeword counters to synchronize the switch to the new FIP setting, and (3) the transmitting transceiver always sends the flag signal.

3. Response to TQ Delta's and Dr. Chrissan's Arguments

23. TQ Delta and Dr. Chrissan agree that the FEC codeword counter embodiment and the flag signal embodiment are different, and that the asserted claims are directed only to the flag signal embodiment. TQ Delta Reply Brief at 9; Chrissan Reply Dec. at ¶ 13. Based on the observation that the FEC codeword counter embodiment and the flag signal embodiment are different, TQ Delta and Dr. Chrissan then conclude that the flag signal cannot contain any information indicating when the updated FIP setting is to be used. TQ Delta Brief at 9 (criticizing Defendants' proposal as "potentially read[ing] on a message that contains information about when the updated settings are to be used"); Chrissan Reply Dec. at ¶¶ 14, 19 (stating that a flag signal is present or not but "does not carry additional information").

24. I disagree for at least four reasons, each of which is discussed in detail below. First, although the Family 6 patents make clear that the FEC codeword counter embodiment is different from the flag signal embodiment, and the asserted claims are directed only to the flag signal embodiment, nothing in the patents would have led a skilled artisan to conclude that the flag signal cannot contain information indicating when the updated FIP setting will be used. On the contrary, and as explained below, given that the transmitter and receiver must synchronize their transitions to a new FIP setting to meet the Family 6 patents' stated objective of avoiding the transceivers having to reinitialize, a skilled artisan would have concluded that the specification does not prohibit the flag signal from containing information to tell the receiver when the transmitter will make the switch to the second FIP setting.

25. Second, a skilled artisan would have understood that with TQ Delta's interpretation of the term "the switching occurs on a pre-defined forward error correction

codeword boundary following the flag signal,” the flag signal would contain information in order for the claimed transmitting transceiver to enable the unclaimed receiver to ensure its switch to the second FIP setting is synchronous with the transmitter’s.

26. Third, a skilled artisan would not have understood the ordinary meaning of the term “flag signal” to preclude the flag signal from containing information.

27. Finally, even if a person having ordinary skill in the art, who would not necessarily have had a background in ADSL, could have connected the terms “sync FLAG” and “inverted sync symbol” used in the Family 6 patents to the portions of G.992.3 identified by Dr. Chrissan, which do not use either the term “sync flag” or “inverted sync symbol,” based on the plain language from the Family 6 patents’ specification, he or she would not have understood the claimed flag signal to be limited to those examples.

a. A Skilled Artisan Would Have Understood That the Flag Signal Can Contain Information To Enable the Transmitter and Receiver To Synchronize Their Transition From the First FIP Setting To the Second FIP Setting.

28. In my opinion, a skilled artisan at the time of the alleged invention would have understood that the flag signal in the Family 6 patents can contain information so that the transmitter and receiver can switch, synchronously, to a new FIP setting. Although I agree with TQ Delta and Dr. Chrissan that the proper construction of the term “flag signal” cannot result in the flag signal embodiment swallowing the FEC codeword counter embodiment, (TQ Delta Reply Brief at 9; Chrissan Reply Dec. at ¶ 13), it does not follow that the flag signal cannot contain any information at all. It simply means that whatever information the flag signal contains is not an FEC codeword counter value. As would have been understood by a person having ordinary skill in the art at the time of the alleged invention, the flag signal can convey

information other than an FEC codeword counter value to enable the transceivers to synchronize the switch to the new FIP setting.

29. The Family 6 patents would have indicated to a skilled artisan that the flag signal sent by the transmitter can contain information that tells the receiver when the transmitter will switch to a new FIP setting. The written description states:

The transmitting modem then sends a flag or marker signal to the receiving modem 200 indicating that the new FIP settings are to be used on a predetermined number of DMT symbols following the transmission of the flag or marker signal. For example, the flag signal could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol. The transmitting modem 300 then starts using the new FIP settings for transmission on the predetermined number of DMT symbols following the transmission of the flag or marker signal. Similarly, the receiving modem starts using the new FIP settings for reception once the predetermined number of DMT symbols following the receipt of the flag or marker signal have been received.

'835 patent at col. 12:25-37. Thus, the written description states that the “flag or marker signal” itself “indicat[es] that the new FIP settings are to be used on a predetermined number of DMT symbols following the transmission of the flag or marker signal.” Furthermore, the Family 6 patents do not explain, in the quoted passage above or elsewhere, how the receiving transceiver would obtain the “predetermined number of DMT symbols” if not from information contained in the flag signal. Likewise, neither TQ Delta nor Dr. Chrissan has explained how the receiving transceiver would obtain the “predetermined number of DMT symbols” if not from the flag signal. Thus, a skilled artisan would have understood that the flag signal can contain information that enables the receiver to identify the “predetermined number of DMT symbols following the transmission of the flag or marker signal” (*e.g.*, the information could be an integer representing the predetermined number of DMT symbols).

30. In explaining FIG. 6, the Family 6 patents state:

Then, in step S630, the transmitter forwards to the receiver a flag signal indicating when the new FIP settings are to be used. At step S640, and at a predefined change time following the transmission of the flag signal, the transmitter begins transmission using the new FIP parameters. Next, at step S650, at the predefined change time following the reception of the flag signal, the receiver commences reception utilizing the new FIP parameters.

'835 patent at col. 19:20-28. This passage also would have indicated to a skilled artisan that the flag signal itself indicates when the new FIP settings are to be used. The Family 6 patents do not explain, here or elsewhere, how the receiver would obtain the “predefined change time” if not from information contained in the flag signal, and neither does TQ Delta or Dr. Chrissan. Thus, a skilled artisan would have understood that the flag signal can contain information enabling the receiver to identify the “predefined change time following the transmission of the flag signal” (*e.g.*, a value representing the time at which the transmitter will switch to the second FIP setting).

31. FIG. 6 itself states, at step S640, after the transmitter has sent to the receiver a “Flag Signal indicating when updated FIP Parameters are to be used” (step S630), that “[a]t the specified change time, Transmitter commences transmission using updated FIP Parameters,” and, at step S650, “[a]t the specified change time, Receiver commences reception using updated FIP Parameters.” The Family 6 patents do not explain how the receiver would obtain the “specified change time” if not from the flag signal sent in step S630. TQ Delta and Dr. Chrissan also fail to explain how the receiver would obtain the “specified change time” if not from the flag signal. Thus, a skilled artisan would have understood that the flag signal can contain information conveying the “specified change time.”

32. A construction of “flag signal” that allows the flag signal to provide information to tell the receiver when to make the switch to the new FIP setting does not cause the asserted claims to extend to the FEC codeword counter embodiment. As a skilled artisan would have

understood, as long as whatever information the flag signal contains is not an FEC codeword counter value, the flag signal embodiment is distinct from the FEC codeword counter embodiment. Furthermore, whereas the FEC codeword counter embodiment requires the transceivers to keep FEC codeword counters, the flag signal embodiment does not, (*see* '835 patent at col. 12:5-7), and therefore whatever information the flag signal contains would not include any FEC codeword counter value in any event.¹

33. According to the written description and drawings, the flag signal embodiment allows the switch to occur (a) “on a predetermined number of DMT symbols following the transmission of the flag or marker signal,” ('835 patent at col. 12:25-37), (b) “at a predetermined change time following the transmission of the flag signal,” ('835 patent at col. 19:20-28), or (c) “at the specified change time,” ('835 patent at FIG. 6). The asserted claims require the switch to “occur[] on a pre-defined forward error correction codeword boundary following the flag signal.” Therefore, aside from the lack of clarity and disclosure on how the “switching” is to occur on a “pre-defined” forward error correction codeword boundary as further discussed below, one having ordinary skill in the art would have understood that the disclosure does not exclude a flag signal that contains information indicating the “predetermined number of DMT symbols,” the “predetermined change time,” or the “specified change time,” any of which could correspond to the claimed “pre-defined forward error correction codeword boundary” on which the switch is to occur, and none of which would result in the flag signal embodiment subsuming the FEC codeword counter embodiment.

¹ I understand the Defendants have proposed the following compromise construction: “signal indicating when updated FIP settings are to be used *and not including a FEC codeword counter value*.” Although I do not think the modification is necessary, I agree that the compromise construction is consistent with the intrinsic evidence for the same reasons discussed herein.

34. From the disclosures of the Family 6 patents, a skilled artisan would have recognized that, contrary to TQ Delta's and Dr. Chrissan's assertions, the flag signal can contain information that enables the receiver to switch to the second FIP setting on the same FEC codeword boundary as the transmitter.

b. TQ Delta's and Dr. Chrissan's Interpretation Of the Term "the switching occurs on a pre-defined forward error correction codeword boundary following the flag signal" Would Have Led a Skilled Artisan to Conclude That the Flag Signal Can Contain Information.

35. Based on how TQ Delta and Dr. Chrissan have interpreted the term "the switching occurs on a pre-defined forward error correction codeword boundary following the flag signal," in my opinion, a skilled artisan would have understood not only that the flag signal can contain information enabling the transmitter and receiver to switch to a new FIP setting, but also that it can contain such information in order for the asserted claims to carry out the Family 6 patents' stated objective to synchronize the transmitter and receiver switching to a new FIP setting to avoid the transceivers having to reinitialize.

36. TQ Delta and Dr. Chrissan contend that the "pre-defined FEC codeword boundary following the flag signal" is any one of the FEC codeword boundaries following the transmission of the flag signal because, they argue, "pre-defined" means only that the positions of the FEC codeword boundaries are "known in advance of the switch." TQ Delta Reply Brief at 32; Chrissan Reply Dec. at ¶ 40. TQ Delta and Dr. Chrissan specifically argue that the switch need not "occur on any one *particular* pre-defined FEC codeword boundary." TQ Delta Reply Brief at 32 (emphasis in original). *See also* Chrissan Reply Dec. at ¶ 40.

37. As the Family 6 patents state repeatedly, the purpose of synchronizing the transceivers' switch to the new FIP setting is to avoid the transceivers having to reinitialize. *See, e.g.,* '835 patent at col. 2:43-3:16; col. 3:37-47; col. 9:19-29. The flag signal embodiment is one

of two disclosed embodiments that synchronize the transceivers' changes to the new FIP setting. '835 patent at col. 11:7-9. Therefore, in the flag signal embodiment, the receiver must switch from the first FIP setting to the second FIP setting at exactly the same point (FEC codeword boundary, DMT symbol, instant in time, etc.) as the transmitter; otherwise, the transition is not synchronized.

38. As a skilled artisan would have understood, if the asserted claims are construed as TQ Delta proposes, the transmitter does not provide to the receiver the information the receiver needs to ensure it switches to the second FIP setting synchronously with the transmitter. TQ Delta argues both that (a) the flag signal does not contain any information that would enable the receiver to ensure that it switches to the second FIP setting at the same time the transmitter does, and (b) the transmitter can switch to the second FIP setting on any one of the FEC codeword boundaries following its transmission of the flag signal. But TQ Delta does not explain how, with these proposed constructions, the receiver knows which FEC codeword boundary the transmitter has selected so that it can switch to the second FIP setting on that same FEC codeword boundary. Consequently, TQ Delta's proposed constructions extend the asserted claims to a transmitter that, from the receiver's perspective, appears to switch, without warning, to using the second FIP setting at some randomly-selected FEC codeword boundary of which the receiver is unaware. As a skilled artisan would have recognized, the result of TQ Delta's proposal for "flag signal," which does not allow the transmitter to provide information in the flag signal to inform the receiver when, exactly, it will switch to using the second FIP setting, is that the receiver would not switch to the second FIP setting at the same time as the transmitter. Consequently, the transceivers would drop the connection and have to reinitialize, which is a scenario the Family 6 patents state that the flag signal embodiment avoids.

39. Although the asserted claims are directed only to the transmitting transceiver and do not include any limitations directed to the receiver, in my opinion, a person having ordinary skill in the art at the time of the alleged invention would still have understood that the proper construction of the disputed terms must carry out the stated objective of the invention, which is to ensure that the transmitter and receiver synchronize their transitions to a new FIP setting to avoid having to reinitialize. In combination, TQ Delta’s proposed constructions of “flag signal” and “the switching occurs on a pre-defined forward error correction codeword boundary following the flag signal” do not require the transmitter to provide to the receiver the information the receiver needs to ensure that its transition to the second FIP setting will be synchronized with the transmitter’s transition. TQ Delta’s interpretation of its proposal for “flag signal” prevents the receiver from obtaining the necessary information from the transmitter and is, therefore, in my opinion, incorrect.

c. A Skilled Artisan Would Have Understood That Flag Signals Can Contain Information.

40. Dr. Chrissan criticizes Defendants’ proposed construction—“signal indicating when updated FIP settings are to be used”—as “remov[ing] the concept of a ‘flag’ being a simple marker that is either present or not present,” even though he acknowledges that Defendants’ proposal uses the exact words used in the Family 6 patents’ specification itself. Chrissan Reply Dec. at ¶ 14. According to Dr. Chrissan, a flag “either occurs or does not occur, but when it occurs it does not carry any additional information within it.” Chrissan Reply Dec. at ¶ 19. Dr. Chrissan then cites various dictionary definitions of the word “flag” to argue that “[t]he word ‘flag’ as used in the term ‘flag signal’ is a term of art that is used to refer to a recognizable signal or bit pattern that serves as a marker for synchronization.” Chrissan Reply Dec. at ¶ 20.

41. Reserved

42. Reserved

43. In my previous declaration, I provided examples of contemporaneous patents that use the term “flag signal” in a manner that comports with Defendants’ construction. Jacobsen Response Dec. at ¶ 46. These patents indicate that Defendants’ construction is consistent with how the term “flag signal” was used in the art at the time of the alleged invention.

44. Moreover, Dr. Chrissan does not provide any convincing explanation as to why a skilled artisan would not understand that a flag signal in the context of the Family 6 patents contains information. Chrissan Reply Dec. at ¶ 14. Neither TQ Delta nor Dr. Chrissan explains how, if not from the flag signal, the receiver obtains the information it needs to ensure its transition to the second FIP setting is synchronous with the transmitter’s transition. As explained above, a skilled artisan would have concluded that the flag signal in the Family 6 patents can contain information that enables the receiver to switch to using the second FIP setting at the same time as the transmitter does, particularly because TQ Delta and Dr. Chrissan contend that the transmitter can switch to the second FIP setting on any one of the FEC codeword boundaries that follows the flag signal.

d. A Skilled Artisan Would Not Have Understood the Claimed “flag signal” To Be Limited To a “sync flag or inverted sync symbol.”

45. As I stated in my previous declaration, it is my opinion that a person having ordinary skill in the art at the time of the alleged invention would not have understood the flag signal of the asserted claims to be limited to a “sync flag or inverted sync symbol,” at least because (1) neither “sync flag” nor “inverted sync symbol” was a term of art, and (2) the Family 6 patents state explicitly that “sync flag” and “inverted sync symbol” are merely examples of “flag signals.”

46. As I stated in my previous declaration, at the time of the alleged invention, neither “inverted sync symbol” nor “sync flag” was a term of art, and neither term appears in the G.992.3 Recommendation in force on the priority date of the Family 6 patents. *See, e.g.*, G.992.3 (07/2002) (defining “sync symbol” but not “inverted sync symbol,” and not defining “sync flag”). The Family 6 patents themselves provide no definition for either “inverted sync symbol” or “sync flag,” and therefore a skilled artisan would not have known exactly what either an “inverted sync symbol” or “sync flag” is.

47. In his reply declaration, Dr. Chrissan argues that “an ‘inverted sync symbol’ and ‘sync flag’ are terms one of skill in the art would understand at least because the Family 6 patents point to the specific portion of the G.992.3 standard that explains what is meant by these terms.” Chrissan Reply Dec. at ¶ 15. According to Dr. Chrissan, a person having ordinary skill in the art at the time of the alleged invention—which Dr. Chrissan believes would be someone with experience in multicarrier modulation but not necessarily ADSL or even DSL generally (Chrissan First Dec. at ¶ 17)—would have done the following:

- (a) Read the sentence in the Family 6 patents, “For example, the flag signal could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol.” ’835 patent at col. 12:29-31.
- (b) Located and sifted through at least 579 pages (as of the ’835 patent’s filing date) or 1,353 pages (as of the ’162 patent’s filing date) (*see* Jacobsen First Dec. at ¶¶ 26-28) of documents included in “ADSL2 G.992.3” and somehow zeroed in on section 8.7.3 of the July 2002 version of G.992.3, even though that section does not use the acronym “OLR,” the term “sync FLAG,” or the term “inverted sync symbol.”

- (c) Read section 8.7.3 and “readily underst[ood] that ‘the phase of the next inserted synchronization symbol shall be inverted’ in G.992.3 describes an inverted sync symbol and that a ‘PMD.Synchflag’ primitive and/or the ‘line signal corresponding to a PMD.Synchflag primitive’ in the G.992.3 standard describes a ‘sync flag.’”

Chrissan Reply Dec. at ¶¶ 15-17.

48. In my opinion, a skilled artisan without experience in ADSL2 would not have been able to make the connections Dr. Chrissan suggests without expending an unreasonable amount of effort to understand the material in G.992.3 and extrapolate from the teachings of G.992.3 to the terminology used in the Family 6 patents. Even Dr. Chrissan needed more than a page in his declaration to make connections he contends would be “readily underst[ood].” If the terms “sync flag” and “inverted sync symbol” were actually terms of art, as Dr. Chrissan contends they were, (Chrissan Reply Dec. at ¶ 15), they would have appeared verbatim in G.992.3. As Dr. Chrissan’s explanation makes clear, neither term appears in G.992.3, neither was a term of art, and neither would have been readily understood by a person having ordinary skill in the art, who would not necessarily have been working in ADSL or had any familiarity with G.992.3.

49. Ironically, TQ Delta criticizes Defendants’ proposed construction as allegedly “making the understanding of the term by a fact finder more difficult,” when Dr. Chrissan required two full pages to explain how a person having ordinary skill in the art would start from the term “flag signal” and end up concluding that a flag signal includes only a “sync flag” and “inverted sync symbol,” two terms that would not have had a definite meaning to a skilled artisan, much less a jury member.

50. Furthermore, and as I stated in my previous declaration, it is unclear from the Family 6 patents, TQ Delta’s opening brief, and Dr. Chrissan’s opening declaration whether there is even any difference between an “inverted sync symbol” and a “sync flag.” *See, e.g.*, TQ Delta Brief at 18; Chrissan Dec. at ¶ 47 (“A flag or marker signal similar to that used in the ADSL2 OLR protocol is *an inverted sync signal, also described as a sync flag.*”). Dr. Chrissan now appears to contend that a “sync flag” is not the same as an “inverted sync symbol,” and the terms are not interchangeable. *See, e.g.*, Chrissan Reply Dec. at ¶ 17 (“a person of skill in the art would readily understand that ‘the phase of the next inserted synchronization symbol shall be inverted’ in G.992.3 describes an inverted sync symbol and that a ‘PMD.Synchflag’ primitive and/or the ‘line signal corresponding to a PMD.Synchflag primitive’ in the G.992.3 standard describes a ‘sync flag.’”). This apparent change in position further underscores that the terms “sync flag” and “inverted sync symbol” were not terms of art and are unclear, even to Dr. Chrissan.

51. Moreover, even if a skilled artisan could somehow have made the connections Dr. Chrissan sets forth to reach the conclusion that the “sync flag” and “inverted sync symbol” are the elements of G.992.3 that Dr. Chrissan goes to great lengths to identify—which would be asking an awful lot of a person having ordinary skill, but not in ADSL—the Family 6 patents state only that the “flag or marker signal” is “similar to that used in the ADSL2 G.992.3 ORL protocol.” ’835 patent at col. 12:1-5 (emphasis added). A skilled artisan would not have understood “similar to” to mean “identical to” or “limited to.” Indeed, a skilled artisan would have considered such an interpretation to be in conflict with the portion of the Family 6 patents that states explicitly that the “inverted sync symbol” and “sync flag” to which TQ Delta now seeks to limit the definition of “flag signal” are merely exemplary: “For example, the flag signal

could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol.”

’835 patent at col. 12:29-31 (emphasis added). Thus, a skilled artisan would not have understood the claimed “flag signal” to be limited to only the elements of G.992.3 identified by Dr.

Chrissan. As I noted in my previous declaration, the inventor himself understood that the term “flag signal” would have had a broader meaning to a skilled artisan than that proposed by TQ Delta. Jacobsen Response Dec. at ¶ 53.

52. Finally, and most importantly, in my opinion, is that even if the claimed “flag signal” were limited to the “inverted sync symbol or sync flag” proposed and interpreted by TQ Delta, the problem described above would still remain: the receiver would not have the information it needs to ensure it switches to the second FIP setting synchronously with the transmitter, thereby avoiding the transceivers having to reinitialize.

53. It remains my opinion that “flag signal” should be construed as “signal indicating when updated FIP settings are to be used” as the Defendants propose to make clear that the flag signal contains information that enables the receiver to ensure its switch to using the second FIP setting is synchronized with the transmitter’s switch.

C. “flag signal” (claims 8 and 9 of the ’162 patent)

54. I disagree with TQ Delta’s proposed construction of the flag signal as being limited to “inverted sync symbol or sync flag” in claims 8 and 9 of the ’162 patent for essentially the same reasons I disagree with its proposed construction of “flag signal” for claims 8 and 10 of the ’835 patent. Specifically, and as explained above, the transmitter and receiver must synchronize their transitions to the second interleaver parameter value to carry out the flag signal embodiment’s stated objective of enabling the transceivers to switch to a new interleaver parameter value without having to reinitialize. Based on this requirement, a skilled artisan would

have concluded that the flag signal contains information to tell the receiver exactly when the transmitter will switch to using the second interleaver parameter value.

55. In addition, TQ Delta's interpretation of the term "the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary following the flag signal" as meaning any one of the FEC codeword boundaries after transmission of the flag signal would have led a skilled artisan to the understanding that the flag signal can contain information in order for the claimed transmitting transceiver to enable the unclaimed receiver to ensure its transition to the second interleaver parameter value is synchronous with the transmitter's transition to the second interleaver parameter value.

56. Furthermore, a skilled artisan would not have understood the ordinary meaning of the term "flag signal" to preclude the flag signal from containing information, as shown by contemporaneous patents and the fact that signals convey information.

57. Moreover, even if a person having ordinary skill in the art could have expended the substantial effort that would have been necessary to connect the terms "sync flag" and "inverted sync symbol" used in the Family 6 patents to the portions of G.992.3 identified by Dr. Chrissan, which do not use either the term "sync flag" or "inverted sync symbol," he or she would not have understood the claimed flag signal to be limited to those examples because the Family 6 patents' specification explicitly states that "sync FLAG" and "inverted sync symbol" are simply examples of flag signals.

58. It remains my opinion that a person having ordinary skill in the art at the time of the alleged invention would have understood the "flag signal" in claims 8 and 9 of the '162 patent to be a signal indicating when an updated interleaver parameter setting is to be used.

D. “interleaver parameter value”

59. Dr. Chrissan and TQ Delta complain that “Defendants’ dropping of the word ‘parameter’ from their construction [of ‘interleaver parameter value’] blurs the meaning of the term.” Chrissan Reply Dec. at ¶ 25. *See also* TQ Delta Reply Brief at 18-19. But neither TQ Delta nor Dr. Chrissan provides any reason to include the word “parameter” in the construction of “interleaver parameter value.” The Family 6 patents do not use the term “interleaver depth parameter” at all. In contrast, they do use the term “interleaver depth” repeatedly. *See, e.g.*, ’835 patent at col. 2:13-14; col. 12:47; col. 12:63; col. 13:8; col. 13:14; col. 13:24; col. 13:51.

60. Furthermore, and as I explained in my previous declaration, the Family 6 patents do not mention any parameter associated with the interleaver *other* than the interleaver depth D. Thus, a person having ordinary skill in the art would have concluded that the “interleaver parameter” is the interleaver depth, and it would have been clear that the term “interleaver parameter value” refers to the numerical value of the interleaver depth.

61. Moreover, TQ Delta’s construction indicates that it also understands that the claimed “interleaver parameter value” is an interleaver depth. Therefore, it is unclear to me what TQ Delta’s concern is.

62. Despite the fact that the Family 6 patents state that “D is the interleaver depth in number of codewords,” (’835 patent at col. 2:13-14), and the Family 6 patents disclose no alternative units for the interleaver depth, TQ Delta faults Defendants’ proposed construction as “improperly” requiring the interleaver depth to be expressed in codewords. TQ Delta Reply Brief at 19. Dr. Chrissan and TQ Delta argue once again that the “interleaver depth need not be expressed in any particular units,” (TQ Delta Reply Brief at 20; Chrissan Reply Dec. at ¶ 26), although Dr. Chrissan later states that “it would be more appropriate to consider interleaver depth as being expressed in a number of bytes.” Chrissan Reply Dec. at ¶ 26.

63. In my opinion, without *some* specified units, the numerical value of the interleaver depth, and, therefore, the “interleaver parameter value,” would have been meaningless to a person having ordinary skill in the art at the time of the alleged invention. The transmitting and receiving transceivers need to know the units in advance in order to communicate properly. The Family 6 patents do not disclose any mechanism by which the units can change during a connection or from one connection to another. “Codewords” is the unit explicitly provided in the specification. Absent knowledge of the units of “interleaver parameter value,” a skilled artisan would not have understood how to avoid infringing the claims, because “interleaver parameter value” could be any value at all in any conceivable units. The only units for the interleaver depth specified in the Family 6 patents are codewords, and, consequently, those are the units a skilled artisan would have understood the interleaver depth to have in the context of the Family 6 patents.

64. Moreover, contrary to Dr. Chrissan’s assertions, (Chrissan Reply Dec. at ¶ 30), Defendants’ proposal does not preclude the interleaver depth from being a fractional value, *e.g.*, one-half of a codeword. Defendants’ proposal requires the units to be codewords, but it does not require the numerical value to be an integer.

65. Consequently, it remains my opinion that a skilled artisan would have understood “interleaver parameter value” to be “the numerical value of the interleaver depth in number of codewords.”

E. “FIP setting”

66. Restating its arguments from its opening brief, TQ Delta argues that “[t]he claims, contrary to Defendants’ proposed claim construction, do not require any particular set of forward error correction and interleaver parameters,” and that they “do not require any particular one of the example set of parameters, nor do they exclude other forward error correction or interleaver

parameters.” TQ Delta Reply Brief at 26. I disagree. First, the Family 6 patents do not disclose the use of any particular type of FEC or interleaver. Therefore, a skilled artisan would not have understood the “FIP setting” to extend beyond the settings of the FIP parameters disclosed in the patents: N, K, R, and D.

67. Second, the Family 6 patents state explicitly that the “FIP parameters” are all of N, K, R, and D—not that N, K, R, and D are examples of FIP parameters, or that some subset of N, K, R, and D constitutes the FIP parameters. *See, e.g.*, ’835 patent at col. 13:43-45 (“While these examples restrict the changes to a subset of the FIP parameters, they can obviously be extended to cover any combination of the FIP parameters (N, K, R and D).”). The Family 6 patents also indicate that the FIP settings are the values of N, K, R, and D. *See, e.g., id.* at col. 3:37-47 (“A plurality of FIP settings can be used for transmission and reception. In accordance with one particular aspect of this invention, the system can transition from one FIP setting to another FIP setting without going through the startup initialization procedure such as the startup initialization sequence utilized in traditional xDSL systems. For example, an xDSL system that implements the systems and methods described herein could start using an FIP setting of (N=255, K=247, R=S, D=64) and then transition to an FIP setting of (N=255, K=239, R=16, D=64) without re-executing the startup initialization procedure.”). The Family 6 patents also disclose that N is the FEC codeword size in bytes (*see, e.g., id.* at col. 2:12 (“N is the codeword size in bytes”)); K is the number of information bytes in a codeword (*see, e.g., id.* at col. 2:16 (“K is the number of information bytes in a codeword”)); R is the number of parity or redundancy bytes in a codeword (*see, e.g., id.* at col. 2:12-13 (“R is the number of parity (or redundancy) bytes in a codeword”)); and D is the interleaver depth in number of codewords (*see, e.g., col. 2:13-14* (“D is the interleaver depth in number of codewords”)).

68. Furthermore, as I explained in my previous declaration, the Family 6 patents always use the term “FIP setting” to refer to all four of the FIP parameters included in Defendants’ construction, even if, in particular examples, fewer than all of the FIP parameters are modified. *See, e.g.*, ’835 patent at 3:42-47 (“For example, an xDSL system that implements the systems and methods described herein could start using an FIP setting of (N=255, K=247, R=S, D=64) and then transition to an FIP setting of (N=255, K=239, R=16, D=64) without re-executing the startup initialization procedure.”); *id.* at col. 12:44-47 (“In this example only the number of information bytes in a codeword (K) and the number of parity bytes in a codeword (R) are updated on-line. The Codeword Size (N) and Interleaver Depth (D) are not changed.”); *id.* at col. 13:4-10 (“This means provided that the modification if [sic] the FIP setting is restricted to K and R, the transition between FIP settings can be done in a seamless manner. This is the case because if the codeword size N and the interleaver depth D are not modified, the transition can happen without the problem of ‘interleaving memory flushing.’”); *id.* at col. 13:21-25 (“In this example only the Codeword Size (N) and the number of parity bytes in a codeword (R) are updated on line. The number of information bytes in a codeword (K) and Interleaver Depth (D) are not changed and therefore the user data rate does not change.”) (emphases added).

Therefore, a skilled artisan would have understood that the FIP parameters are all of N, K, R, and D, and that the “FIP setting” is “forward error correction and interleaver parameters characterized by the set of parameters for codeword size in bytes, number of information bytes in a codeword, number of parity or redundancy bytes in a codeword, and interleaver depth in number of codewords.”

69. TQ Delta and Dr. Chrissan criticize Defendants’ construction for not including the word “value.” TQ Delta Reply Brief at 27; Chrissan Reply Dec. at ¶ 34. TQ Delta and Dr.

Chrissan complain that “[w]ithout the word ‘value’ in the construction, the change from a first FIP setting to a second FIP setting could be confused as requiring a switch from a first set of parameters, e.g., parameters A and B, to a different set of parameters, e.g., parameters B and C.” TQ Delta Reply Brief at 27; Chrissan Reply Dec. at ¶ 34. This argument is incorrect because Defendants’ construction includes all four of the FIP parameters N, K, R, and D and, like the Family 6 patents, does not exclude situations in which fewer than all of the parameter values change from the first FIP setting to the second FIP setting. Thus, Defendants’ construction would never result in a switch from a first set of parameters to a different set of parameters.

70. TQ Delta also argues that the construction of “FIP setting” should not specify units for the FIP parameters N, K, R, and D. TQ Delta Reply Brief at 28. I disagree. Although the Family 6 patents describe the FIP settings in the context of DSL, there is no indication that the units of these settings’ values (*i.e.*, N, K, and R in bytes, and D in codewords) would be different for other kinds of systems. For example, the Family 6 patents do not provide any non-DSL examples of FIP settings. Moreover, without *some* units for each of the FIP parameters, the parameters N, K, R, and D would have been meaningless to a person having ordinary skill in the art at the time of the alleged invention. The only units specified for N, K, R, and D in the Family 6 patents are those included in Defendants’ construction.

F. “FIP value”

71. It remains my opinion that a skilled artisan would have understood “FIP value” to be the “numerical value of codeword size in bytes, number of information bytes in a codeword, number of parity or redundancy bytes in a codeword, or interleaver depth in number of codewords” as Defendants’ propose. The Family 6 patents do not define the term “FIP value” or provide examples of “FIP values,” but a person having ordinary skill in the art would recognize based on the disclosures of the Family 6 patents that a “FIP value” is simply the value of one of

the “FIP parameters.” As explained above, the “FIP parameters” are N, K, R, and D. *See, e.g.*, ’835 patent at col. 13:43-45 (“While these examples restrict the changes to a subset of the FIP parameters, they can obviously be extended to cover any combination of the FIP parameters (N, K, R and D).”).

G. “the switching occurs on a pre-defined forward error correction codeword boundary”

72. TQ Delta and Dr. Chrissan previously argued that “pre-defined” in the context of this term “means that both the transceiver [*sic*, transmitter] and receiver independently know how to calculate an exact FEC codeword boundary transition point relative to an occurrence of the sync flag, based on a protocol defined by a standard and/or by the designer of the transceiver devices.” TQ Delta’s Opening Brief at 24 (emphasis added). They also argued that because transceivers implementing “the incorporated DSL standards” “know” the positions of the FEC codeword boundaries, the term “the switching occurs on a pre-defined forward error correction codeword boundary” is definite. TQ Delta’s Opening Brief at 24; Chrissan First Dec. at ¶ 59.

73. In my previous declaration, I noted that the claims do not appear to require the use of any protocol or standard, and TQ Delta and Dr. Chrissan do not appear to contend that a standard is necessary to practice the claims. Jacobsen Response Dec. at ¶ 86. Furthermore, I observed that the Family 6 patents do not disclose where the pre-defined FEC codeword boundary comes from (*e.g.*, who defines it, where or how it is defined, etc.) or what its value might be. Jacobsen Response Dec. at ¶ 90. Perhaps realizing that the Family 6 patents do not disclose the use of or information about any protocol or standard required to implement the claimed apparatus or the flag signal embodiment, Dr. Chrissan and TQ Delta now contend that the term “the switching occurs on a pre-defined forward error correction codeword boundary” means only that the transmitter switches to the second FIP setting on any one of the many FEC

codeword boundaries that follow its transmission of the flag signal. TQ Delta Reply Brief at 32; Chrissan Reply Dec. at ¶ 40.

74. I disagree. In my opinion, a skilled artisan would not have understood “pre-defined” to mean any codeword boundary selected from the set of FEC codeword boundaries that are known in advance of the switch and that follow the transmission of the flag signal, as TQ Delta and Dr. Chrissan contend. TQ Delta Reply Brief at 32; Chrissan Reply Dec. at ¶ 40. A skilled artisan would have understood that whenever a transmitter and receiver communicate using forward error correction “during steady-state communication or initialization,” as the asserted claims require them to do, the transmitter and receiver always know the locations of all of the FEC codeword boundaries. Specifically, the transmitter knows where all of the FEC codeword boundaries are because it generates the FEC codewords, and the receiver knows where all of the FEC codeword boundaries are because it decodes the FEC codewords; otherwise, the transmitter and receiver would not be communicating “during steady-state . . . or initialization” as the preamble of the asserted claims requires. Therefore, in my opinion, TQ Delta’s interpretation of “pre-defined forward error correction codeword boundary” reads the word “pre-defined” out of the claim term because, by definition, all FEC codeword boundaries are always known to a transmitter and receiver that are communicating “during steady-state . . . or initialization” using FEC.

75. The Family 6 patents confirm that “pre-defined” means more than simply that the codeword boundary is any one of all of the known FEC codeword boundaries occurring after transmission of the flag signal but before the switch. For example, FIG. 6 illustrates the flag signal embodiment. ’835 patent at col. 19:15-16. At step S630, the transmitter forwards to the receiver a “Flag Signal indicating when updated FIP Parameters are to be used.” The very next

step is S640, which says, “At the specified change time, Transmitter commences transmission using update FIP Parameters.” The following step, S650, uses similar words: “At the specified change time, Receiver commences reception using updated FIP Parameters.” From the wording of steps S640 and S650, a skilled artisan would have concluded that the transmitter’s change to the second FIP setting in claim 8 does not occur on just any FEC codeword boundary following its transmission of the flag signal, but rather on the one that coincides with “the specified change time.”

76. Elsewhere, the written description states:

The transmitting modem then sends a flag or marker signal to the receiving modem 200 indicating that the new FIP settings are to be used on a predetermined number of DMT symbols following the transmission of the flag or marker signal. For example, the flag signal could be an inverted sync symbol, or sync FLAG, as used in the ADSL2 G.992.3 OLR protocol. The transmitting modem 300 then starts using the new FIP settings for transmission on the predetermined number of DMT symbols following the transmission of the flag or marker signal. Similarly, the receiving modem starts using the new FIP settings for reception once the predetermined number of DMT symbols following the receipt of the flag or marker signal have been received.

’835 patent at col. 12:25-37. From this disclosure, a skilled artisan would have concluded that the “pre-defined FEC codeword boundary” of claim 8 is the one corresponding to the “predetermined number of DMT symbols.” But, again, a skilled artisan would not have concluded that the “pre-defined FEC codeword boundary” could be just any one of the FEC codeword boundaries following the flag signal, all of which are “pre-defined” in TQ Delta’s view.

77. Consequently, in my opinion, “pre-defined” does not simply mean that “the position of the boundary of each codeword is known prior to the switching” as TQ Delta contends. As I explained in my previous declaration, in order for the receiver to synchronize its

use of the second FIP setting with the transmitter's use of the second FIP setting, it is necessary for the receiver to know not only where the FEC codeword boundaries are, but also which of those boundaries is the claimed "pre-defined" one. Just knowing where each of the FEC codeword boundaries is, which is all that TQ Delta's construction requires, would not enable a person having ordinary skill in the art, or the receiver, to determine where the "pre-defined forward error correction codeword boundary" on which the switch will occur is.

78. As explained above in the discussion of "flag signal," TQ Delta's proposed construction of "the switching occurs on a pre-defined forward error correction codeword boundary" coupled with its proposed construction of "flag signal" does not guarantee that the claimed apparatus, in communication with a receiver, would carry out the flag signal embodiment. Specifically, because, according to TQ Delta, the flag signal cannot contain any information at all, and the transmitter can switch from a first FIP setting to a second FIP setting on any one of the FEC codeword boundaries that follows transmission of the flag signal, the receiver has no way to ensure that its transition to the second FIP setting is synchronous with the transmitter's transition.

79. Particularly in light of TQ Delta's interpretation of the "pre-defined forward error correction codeword boundary" as any of the known FEC codeword boundaries that follow the flag signal but precede the transmitter's transition to the second FIP setting, it remains my opinion that a person having ordinary skill in the art would not have been able to identify the "pre-defined forward error correction codeword boundary" with any reasonable certainty. Consequently, a skilled artisan would not have understood the term "the switching occurs on a pre-defined forward error correction codeword boundary" or how to avoid infringing the asserted claims.

H. “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary”

80. In my opinion, and for the reasons given in the previous section, it remains my opinion that a person having ordinary skill in the art at the time of the alleged invention would not have known how to interpret the term “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary” because he or she would not have known with reasonable certainty what the “pre-defined forward error correction codeword boundary” would be. Although the written description of the Family 6 patents states that the purpose of the flag signal is to ensure that the transmitter and receiver synchronize their transitions to new a new FIP setting, TQ Delta’s proposed constructions for the terms “flag signal” and “the second interleaver parameter value is used for transmission on a pre-defined forward error correction codeword boundary” do not ensure that the receiver has the information it needs to transition to the new FIP setting at the same time the transmitter does.

I declare under penalty of perjury of the laws of the United States that the foregoing is true and correct.

Date: December 18, 2017



Krista S. Jacobsen